Commodore

Cena 10 tys. zł
nr indeksu 355275

4'92

KEDAD

Miesięcznik Użytkowników Komputerów C-64 i Amiga



Commodore



nr indeksu: 355275

Wydawca:
KEBAB - sp. z o.o.
ul.Wojciechowskiego 28
PL - 71 476 Szczecin
telefon: (091) 77674
telefax: (091) 45402

Redaguje kolegium w składzie: Krzysztof Kobus, Patryk Łogiewa, Grzegorz Mikuła, Krzysztof Moroń, Paweł Sołtysiński.

Prezes zarządu Spółki:
Piotr Sołtysiński

Szef działu AMIGA:
Krzysztof Kobus
tel.:(091)525336

Szef działu C-64:
Paweł Sołtysiński
tel.:(091)77674

Stale obecni na łamach:
Robert Turliński
Miłosław Smyk
Arkadiusz Zych

Redakcja nie zwraca nie zamówionych materiałów, oraz zastrzega sobie prawo wprowadzania zmian w otrzymanych rękopisach.

Projekt okładki:
Dariusz Zawadzki

Zdjęcia:
Sławomir Borek /"Panorama"

W sprawie kolportażu...

I TY MOŻESZ BYĆ KOLPORTEREM "KEBABA"!

Chcielibyśmy zainteresować naszą propozycją te osoby fizyczne i prawne, które chciałyby się podjąć kolportażu we własnym zakresie. Zapewnimy jednocześnie co najmniej tygodniowe wyprzedzenie przed przekazaniem nakładu do dystrybucji przez krajowego kolportera. Zapraszamy do współpracy studia komputerowe, księgarnie i obrotne osoby indywidualne. Szczegółowych informacji idziela się telefonicznie i w siedzibie redakcji.

... reklamy ...

Przedsiębiorstwo KEBAB spółka z o.o. oferuje Państwu szybką i tanią obsługę reklamową. Ogłoszenia drobne od osób indywidualnych (do 10 słów na wyciętym z numeru kuponie) przyjmujemy bezpłatnie. Większe - 1000 zł za słowo. Reklamy ramkowe (minimalny format - 20 cm kwadratowych): 1 cm2 - 4500 zł; cała strona - 2,5 miliona zł; dodatkowy kolor - odpowiednio 50% drożej. Treść ogłoszeń przyjmujemy za pośrednictwem poczty (adres - patrz stopka redakcyjna) lub Agencji Informacyjnej "SIEĆ"; Warszawa, ul.Prezydencka 11; tel.255-433; fax.254-164. Ogłoszenia wraz z określeniem formatu reklamy prosimy nadsyłać listem poleconym. Dołączenie odcinka wpłaty znacznie przyspieszy zamieszczenie ogłoszenia.

... i prenumeraty.

Aby uporać się z problemem ciągłego wzrostu cen usług poligraficznych, papieru itp. i uniknąć dokonywania przez Czytelników Kłopotliwych dopłat, postanowiliśmy wprowadzić tzw. małą prenumeratę w okresach 3-miesięcznych. Rozwiązanie to gwarantuje Czytelnikom niezmienność ceny w okresie, który obejmuje zamówienie. Cenę egzemplarza wraz z kosztem usługi pocztowej skalkulowaliśmy na 9500 zł. Daje to następujące możliwości:

numery 1, 2/3, 4, 5	- 38000,-
numery 2/3, 4, 5	- 28500,-
numery 4, 5	- 19000,-

Kwoty te (zależnie od zamówienia) należy wpłacać na konto:

KEBAB sp. z o.o.
Pomorski Bank Kredytowy II Oddział w Szczecinie
konto nr: 368113-25771-136

Należy również podać DOKŁADNY ADRES, IMIĘ I NAZWISKO zamawiającego!



Nr 4 kwiecień 1992

AMIGA 600
opis najnowszej wersji Amigi znajdziecie na stronie 20

Tape - Burger
to kolejny kopier do "wklepania" na C-64 tym razem specjalnie dla użytkowników kaset

Trzecie kroki z nowym nabytkiem
tzn. zaczynamy programować!

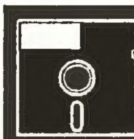
Assembler na C-64
dalszy ciąg kursu na stronie 12.

Koniec z "miganiem"
Flicker-Fixer to spowoduje.

Lubisz pograć ?
a nie potrafisz "przejść" Hudson Hawk'a Czytaj od strony 24.

Spis treści:

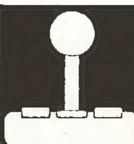












Z kraju i ze świata...

W poprzednim numerze "KEBAB-'a" wspominaliśmy o przystąpieniu dwóch członków grupy Anarchy do tymczasowej współpracy z Core Design. Na rezultaty nie trzeba było czekać długo - jest już w sprzedaży najnowsza gra tej firmy - "Wolfchild", do której muzykę skomponował Nuke, a techniczną stroną, znakomitego nawiasem mówiąc, intra zajął się Dan. Oto więc kolejny z wielu przypadków przechodzenia postaci ze sceny na profesjonalny, dający możliwość podpreperowania budżetu, rynek.

To, że Amiga nadaje się doskonale do zastosowań związanych z animacją komputerową wie prawie każdy. Niestety, aby uzyskać szybką, płynną animację potrzeba na ogół dużej pamięci operacyjnej, posiadaniem której już nie wszyscy mogą się pochwalić. Z myślą o użytkownikach takich właśnie ubogich w RAM Amig grupa (tak tak!) DIANETIX przygotowuje taśmę wideo zawierającą, oprócz animacji wykonanych przez profesjonalistów, również programy demonstracyjne, które ani myślą działać na standardowych, czy nawet wyposażonych w 1MEG Amigach. Kaseta ma być dostępna u VHS-swapper'ów w pierwszej połowie tego roku.

Największa, najpopularniejsza i chyba najrealniej odzwierciedlająca stan faktyczny lista rankingowa amigowskiej sceny - "Eurocharts" - przestała istnieć. U podstaw decyzji o likwidacji leżało prawdopodobnie znużenie członków grupy Crusaders, którzy oprócz podliczania setek głosów z całej Europy mieli na głowach przygotowanie publikacji od strony technicznej. Głosowanie przeprowadzano w następujących kategoriach: megademo, demo plikowe, grupa, koder, grafik, muzyk, magazyn dyskowy, gra zręcznościowa, gra strategiczna, demo pack, BBS, film i LP. Dwie ostatnie pozycje wykraczają nieco poza tematy czysto komputerowe, pozwalały one jednakże zorientować się w gustach przeciętnego Amigowca. Mimo iż grupa Crusaders nie szczędziła wysiłków by notowania były jak najbardziej fair, to jednak od czasu do czasu pojawiały się głosy zarzucające redaktorom stronniczość, przekupstwo i poddające tym samym w wątpliwość sens całego przedsięwzięcia. Niemniej, "Eurocharts" przez ogromną większość, tak zwykłych użytkowników jak i osób działających na scenie, był uważany za miarodajne źródło informacji i głos rozstrzygający we wszelkiego rodzaju sporach "lepsze nasze a nie wasze". Teraz, gdy "Eurocharts" jest już przeszłością, należałoby zastanowić się w jaki sposób zapełnić powstałą lukę. Ponieważ wielcy komputerowego światka mają ręce i tak pełne roboty z produkcją programów demonstracyjnych, gazetek i całej reszty będącej potem przedmiotem głosowania, propozycja stworzenia nowej listy rankingowej skierowana jest raczej do jakiejś stosunkowo młodej, prężnej, niekoniecznie znanej grupy. Jeśli zamierzenie byłoby następnie realizowane na poziomie co najmniej dorównującym pierwowzorowi, to owa grupa w krótkim czasie miałaby szansę znaleźć się na amigowym Olimpie. Ze swej strony, jako redakcja czasopisma blisko związanego ze zbyt wieloma grupami by pozwolić sobie na stronniczość względem którejś z nich, proponujemy Czytelnikom zorganizowanie całkowicie obiektywnych chart'ów na naszych łamach. Forma, kategorie głosowania, uwzględnienie sceny C64, ewentualny podział na Polskę i resztę świata - to wszystko zależy od Was. Sugestie i propozycje na temat kształtu "Kebabowych czartów" prosimy więc nadsyłać na adres:

Miłosław Smyk,
Orawska 22/34,
70-131 Szczecin.
z dopiskiem "Diabla Lista".

Dwaj członkowie węgierskiej grupy FACES pracują aktualnie nad skończeniem nowej gry logicznej dla Amigi i C64 o nazwie „ILLOGICAL". Rozmowy z ewentualnymi wydawcami trwają.

Popularny magazyn dyskowy dla C64 o nazwie „MAMBA" przestał się ukazywać. Dotychczas jego wydawaniem w języku angielskim zajmowała się grupa ENIGMA (w przeszłości CRAZY).

Znana z wielu dem grupa BANZAI rozpadła się. Obecnie większa część składu osobowego tej grupy postanowiła reaktywować STARION.

Nowe demo holenderskiej grupy BLACKMAIL było bardzo długo zapowiadane i nie zawiodło oczekiwań (patrz PUBLIC DOMAIN dla C64). Nie wszyscy jednak wiedzą, że poprzedzone zostało ono przez pewien żart: inne demo, któremu nadano tę samą nazwę - Dutch Breeze. Zamieszania było mnóstwo i oto chyba autorom „falstartu" chodziło...

Według magazynu dyskowego BRUTAL RECALL dziesiątka najlepiej piszących dema grup w Europie przedstawia się następująco:

1. Crest
2. Flash Inc.
3. Bonzai (już nie istnieje)
4. Light
5. Censor Design
6. Cosmos Design
7. Faces
8. Triad
9. BlackMail
10. Fairlight

Rune-Gram Madsen, czyli wspominany często autor znakomitego ASM-One'a zakończył 10 marca prace nad kolejną wersją wspomnianego wyżej assemblera. Według zapewnień autora Nowy ASM-One w wersji 1.21 jest jeszcze szybszy (?!) od swojego poprzednika i nie posiada już kilku bug'ów które ciążyły na wersji 1.01.

Po sukcesach targów Amiga 91 w Berlinie i w Kolonii, Firma Commodore Business Machines (tak brzmi nazwa tej firmy) postanowiła w tym roku uruchomić swoje własne targi. Oprócz imprez serii "Amiga", odbędą się w tym roku w dniach 26-29 listopada targi pod nazwą "World of Commodore 1992". Siedzibą targów będzie hala nr 5 terenów targowych we Frankfurcie nad Menem.

W minionym roku dużym powodzeniem klientów cieszyły się różne wersje Flicker-Fixer'ów. Przebojem stał się opisywany w tym numerze Multivision, głównie ze względu na niską cenę. W tym roku nadszedł czas na grafikę 24-o bitową. Nic dziwnego, w końcu już nawet "pecety" z odpowiednią kartą przewyższają możliwościami graficznymi standardowe 4096 kolorów Amigi. W odpowiedzi pojawiły się w tym roku urządzenia zapewniające Amidze dostęp do 16,8 miliona kolorów. Harlequin, GVP - Impact Vision, FrameMaster, DCTV, ColorMaster 24 to niektóre tylko z pozycji hardware'owych dotyczącej tej szybko się rozwijającej dziedziny. Odpowiednia karta graficzna w połączeniu z dobrym programem do obsługi 24 bitplane'ów potrafi czynić cuda o jakich nam się jeszcze niedawno nie śniło. Niestety gorzką pigułką pozostają jak zwykle ceny: średnio 4000,- DM za kartę i 3000,- DM software. Najtańsze rozwiązania nie kosztują poniżej 1000,- DM.

Wzmiankowany ostatnio program do Ray-Tracingu: "Reflections 2.0" cieszy się coraz większym powodzeniem na rynku europejskim. Z pewnością stosunkowo niewysoka (jak na tego typu pakiet) cena jest obok sporych możliwości głównym elementem ułatwiającym wybór ewentualnemu nabywcy. Niemniej jednak producenci nie spoczywają na laurach i wypuszczają coraz nowe elementy uatrakcyjniające dodatkowo program. Na razie dostępne są dyskietki zawierające gotowe objekty w następujących grupach tematycznych:

- science - fiction
- owady
- meble

W przygotowaniu są już fonty oraz tzw. textures (miłośnicy ray-tracingu na pewno znają to słowo)

Targi, targi, targi... tak! Namnożyło nam się ostatnio rozmaitych targowisk. Najpierw CEBIT 92, potem Amiga 92 no i nasz rodzimy Infosystem 92. O ile ostatnie targowisko (poznańskie) stosunkowo mało nas interesuje ze względu na brak wystawców prezentujących tak bardzo nas interesujące nowinki i ciekawostki dotyczące komputerów Commodore, o tyle zarówno na targach CEBIT jak i Amiga 92 można było sporo zobaczyć. Do sensacyjnych niespodzianek należy chyba zaliczyć prezentację przez firmę Commodore nowej Amigi (600). W ogóle trzeba przyznać, że po ubiegłorocznych sukcesach finansowych (w dużym stopniu dzięki rynkowi polskiemu oraz byłej NRD), firma stała się zdecydowanie bardziej dynamiczna. Co targi, to coś nowego! Najpierw CDTV, potem 500plus no a teraz "sześćsetka". Tylko czekać co pokażą pod koniec roku w Kolonii. Niestety przedstawiciele firmy byli widać tak zmęczeni prezentacjami swojego nowego produktu, że postanowili odpocząć i na targach Amiga 92 w Berlinie nie pokazali się wcale. Trzeba stwierdzić, że brak stoisk Commodore, które w ubiegłym roku zajmowały lwią część powierzchni wystawowej wpłynął zdecydowanie na obniżenie rangi imprezy berlińskiej. Niemniej jednak nie zawiedli główni producenci i dystrybutorzy osprzętu i oprogramowania Amigi. Supra, GVP, FSE, Rossmoeller, Vortex, Kolff Computer Supplies (KCS), Alfa Data, Masoboshi to tylko niektóre (i starsze) z firm prezentujących nowe (i starsze) produkty do Amigi. Do najciekawszych rzeczy należy chyba zaliczyć najnowsze turbo-karty z procesorem 68040 firm GVP oraz PP&S, karty emulujące serię 386SX komputerów PC oraz ATonce plus (Vortex), 12-o bitowe karty dźwiękowe oraz 24-o bitowe (16,8 mln kolorów) karty graficzne. Z tańszych rzeczy można było pooglądać, podotykać a nawet kupić np. Videobackup czyli nowe, rewelacyjne rozwiązanie problemu zabezpieczania zawartości twardego dysku przy użyciu dowolnego magnetowidu. Także stacje dysków o pojemności do 1,64 MB, pracujące z dyskietkami HD. Sporo było również oprogramowania od Public Domain począwszy a na DynaCaDD czy TVPaint (pakiet graficzny do obsługi grafiki 24-o bitowej) kosztujących po 2950,- DM kończąc. Był już również wspominany w poprzednim KEBAB'ie FontDesigner w pełnej wersji, jak również programy typu ExpertDraw, czy ProVector (wektorowe tworzenie grafiki). Caligari V2.1, Reflections 2.0 czy Real 3D V1.4 kusiły fanów Ray-Tracingu Programy typu DirOpus czy Diskmaster w najnowszej wersji konkurowały ze sobą (na różnych stoiskach) cenowo. Ogólnie mimo, jak już wspomniałem znacznie mniejszej, w porównaniu z ubiegłym rokiem rangi imprezy, warto było popatrzeć.

64

Piractwo !

ciemne i jasne strony zjawiska.

Kilka lat temu w większych miastach naszego kraju zaczęły powstawać dziwne, ale bardzo pożyteczne dla nas twory - giełdy komputerowe. Głód programowy jaki odczuwali liczni już użytkownicy 8-bitowego sprzętu zmusił ich do bardzo czynnego działania, a i że pieniądze mogły z tego skapnąć to interes zaczął się bardzo dynamicznie rozwijać. Po kilku latach przybrał on takie wymiary, że przebywający tam przedstawiciele zachodnich firm software'owych mdleli lub nawet dostawali ataku serca, gdy miły chłopczyk oferował im ich produkt za bardzo korzystną cenę. Takich przypadków było wiele i nie ma sensu tu ich wszystkich przytaczać. Poprzez ten list chciałbym, aby każdy Czytelnik (nawet ten nie związany z tematem) mógł wyrobić swoje, własne zdanie na ten drażniący wielu ludzi temat.
Cała Europa Zachodnia od wielu lat przestrzega prawo, które mówi, że osoba, która dopuściła się kradzieży produktu lub patentu danej firmy podlega karze do 10 lat więzienia, lub grzywny równej sumie wszystkich poniesionych start z tytułu kradzieży mienia. Prawo to jest respektowane od Odry na zachód i nosi nazwę prawa autorskiego. Wszyscy bardzo chcemy, aby zachodni kapitał jak najszybciej wpłynął do Polski. Niestety pozostałości dawnego systemu i wiele innych jeszcze nie uregulowanych spraw stanowią poważną barierę. W takim chaosie bardzo często tworzą się różnego rodzaju organizacje przestępcze, działające pod przykrywką jakiejś spółki, które to działają na niekorzyść firm zachodnich, a pośrednio na niekorzyść naszego społeczeństwa. Z racji naszego zainteresowania (komputery, oczywiście) bardzo byśmy pragneli, aby takie smoki jak IBM, Commodore, Apple, czy Atari wkroczyły do naszych miast, do pięknych sklepów z miłą obsługą, no i oczywiście z dobrym i tanim towarem. Ale zaraz, chwileczkę, każdy komputerowiec wie (giercman w szczególności), że komputer bez jakiegokolwiek oprogramowania jest tylko drogocennym złomem. Tak więc marzymy o takich firmach software'owych jak MICROSOFT, MANDARIN, czy PSYGNOSIS. Cóż, możemy tak sobie marzyć i przez lata, ale jeśli nadal w Polsce nie wyjdą stosowne ustawy to nadal pozostaniemy jedynie na etapie marzeń. Co więc zostaje nam do zrobienia? Niestety, ale jedynie czekać i wydaje mi się, że przez dłuższy czas. Jeśli takie firmy nie wejdą w ciągu 2 lat - co jest bardzo prawdopodobne - to możemy być pewni, że zostaniemy w technologii komputerowej zepchnięci na jeszcze dalszą pozycję, a jeśli powstanie naprawdę dobry program polskiego autora, to nie miałby on takiej siły przebicia, aby dostać się na bogaty rynek zachodni, ponieważ pochodziłby z państwa bezprawia. W moim rodzinnym mieście przeprowadziłem sondę na temat praw autorskich, czyli jak pewne osoby, instytucje patrzą na problem piractwa komputerowego (i nie tylko) w Polsce. Otóż we wszystkich sklepach typu Audio-Video, w których sprzedawane są komputery różnego typu, panowało przekonanie, że to zjawisko powinno być jak najszybciej wyeliminowane (przecież zdążamy do Europy), ale na moje pytanie co na półkach robią zestawy z grami komputerowymi oraz pirackimi kasetami magnetofonowymi, sprzedawcy robili głupie miny lub uciekali na zaplecze.
Po przeczytaniu tego listu na pewno znajdą się zwolennicy jak i przeciwnicy ustawy o ochronie praw autorskich. Muszę powiedzieć, że istnieje jedna bardzo ważna sprawa (tutaj zgadzam się z piratami), która może być ogromnym problemem w legalizacji prawa autorskiego. Sprawą tą jest przyszła cena



oprogramowania. Każdy wie, że na zachodzie gry są bardzo drogie, a programy użytkowe w przeliczeniu na złotówki stanowią równowartość dobrego, nowego komputera! Narzuca się więc pytanie: czy zachodnie firmy software'owe narzucą nam, aż tak ogromną cenę za swoje produkty? Oczywiście tak, bo czemu mieliby traktować nas inaczej niż innych swoich partnerów. Dlatego też mało kogo będzie stać na kupno nowej gry nie mówiąc już o programach graficznych, muzycznych (na szczęście tutaj istnieją programy typu PD), menadżerskich i innych. W wyniku tego nasze przedsiębiorstwa zostaną w tyle, za zachodnimi, a szary gracz będzie przez pół roku kończyć jedną grę, bo w tym czasie być może uzbiera na drugą. Pieniądze dla nas wszystkich, szczególnie w tych czasach, znaczą wiele. Tak więc co zrobimy, gdy będziemy mieli do wyboru program firmowy, bardzo drogi, ale z estetyczną instrukcją oraz opakowaniem i ten sam program piracki, często zawieszający się, bez żadnych instrukcji (strata czasu na rozgryzanie funkcji programu), ale za relatywnie bardzo małą ilość pieniędzy? Prawie wszyscy wybraliby ten drugi i ja zrobiłbym to samo. Tak samo postąpiliby komputerowcy o niezbyt dużych zasobach pieniężnych. Kończąc powiem, że giełda, czyli piraci przez swoją działalność mogą dać niezbyt bogatym "odrobinę luksusu", przez ich właśnie działalność stosunkowo szybko możemy na swoim monitorze oglądać najnowszy wyrób dobrej firmy, ale również przez nich, być może nigdy nie powstanie rodzimy rynek informatyczny z prawdziwego zdarzenia.

Czekam na listy dotyczące tego tematu.

Piszcie na adres:

Adam Starzak,
ul.Fałata 17/26,
75-427 Koszalin.

Cartridge? Czy nie Cartridge?

X-Power 500 Proffessional

W kolejnym odcinku naszego przeglądu magicznych pudełek proponuję konkurencyjny niejako dla opisywanego ostatnio Action Replay'a cartridge "X-Power" w wersji dla Amigi. Cartridge ten posiada w praktyce bardzo podobne możliwości jak i "Action", lecz producenci postawili na nieco inną koncepcję. Jeżeli w Action Replay'u mamy do czynienia ze stosunkowo ładnie wyglądającą i dopracowaną konstrukcyjnie obudową a bardzo spartańsko przedstawiającym się oprogramowaniem. O tyle w przypadku X-Power'a sprawa ma się dokładnie odwrotnie. Oprogramowanie zostało bardzo dokładnie dopracowane pod względem funkcjonalnym i estetycznym, natomiast obudowa wbrew napisowi "proffessional" wcale takiego wrażenia nie daje. Widać uznano, że w końcu nie to jest najważniejsze. Cartridge kupuje się w standardowym opakowaniu... od kasety video. Instrukcję otrzymujemy owszem, ale na dyskietce. Istnieje co prawda możliwość odczytania instrukcji po uaktywnieniu cartridge'a tzn. nie musimy za każdym razem wykonywać reset'u i boot'owáć (ale świetny wyraz!) komputera przy użyciu załączonej dyskietki.

Tak wygląda X-Power 500 z zewnątrz...



Niemniej jednak rozwiązanie to nie wydaje mi się być najlepszym w porównaniu do normalnej, drukowanej książeczki. Być może twórcy uznali, że oprogramowanie cartridge'a nie wymaga dodatkowych objaśnień. W dużym stopniu jest to prawdą. W przeciwieństwie do pierwszych egzemplarzy AR-III (Action Replay Mk-III) X-Power nie ma żadnych problemów z Kickstart'em 2.0. Po uruchomieniu freezer'a przyciskiem oznaczonym dla niepoznaki napisem "ACTION" pojawia się tabela w której po chwili możemy przeczytać jaką aktualnie dysponujemy wersją kickstart'u, ile (i w jakim obszarze) mamy pamięci CHIP oraz FAST, ile mamy stacji dysków, czy mamy podłaczoną drukarkę itp. itd. O ile AR po wciśnięciu przycisku zostawiał użytkownika sam na sam z kursorem, o tyle wszystkie funkcje X-Power'a wybierane są komfortowo z menu przy użyciu zarówno myszy jak i (na szczęście) klawiatury. Mamy tu praktycznie wzystko co potrzeba ze sławetnym programem X-Copy włącznie. Jeżeli znajdziemy się w sytuacji gdzie nie wiemy co robić to należy spróbować wcisnąć klawisz HELP. Zazwyczaj pojawi się podpowiedź. A gdyby i to było mało to możemy jeszcze wczytać odpowiedni fragment instrukcji z załączonej dyskietki. Do nieprzeciętnych należy zaliczyć szybkość pracy zawartego w systemie cartridge'a monitora pamięci. Również pozostałe narzędzia działają szybko i sprawnie. Ogólne wrażenie jakie pozostawia oprogramowanie X-Power'a jest bardzo pozytywne. Konstrukcja samego cartridge'a (z wyjątkiem obudowy) również nie pozostawia wiele do życzenia. Bardzo istotną cechą która odróżnia X-Power'a na plus od pozostałych cartridge'ów jest jego "przelotowość" to znaczy, że po podłączeniu cartridge'a (i zrezygnowaniu z obudowy) możemy podłączyć do niego dalsze urządzenia przystosowane do pracy z expansion port'em Amigi. Niestety nie zawsze i nie z każdym urządzeniem, połączenie takie zdaje egzamin. Zdarza się, że np. z jednym typem kontrolera twardego dysku musi być podłączony najpierw cartridge a dopiero do niego kontroler a z innym typem trzeba postąpić na odwrót. Jeżeli natomiast kontroler nie jest "przelotowy" to często pozostaje nam stwierdzić, że zestaw taki po prostu nie działa. Jak zatem odpowiedzieć na pytanie: Który cartridge mam kupić (zakładając że jestem zdecydowany na taki wydatek)? Ponieważ żaden z kandydatów (AR-III i X-Power) nie posiada rażących wad a zakres ich zastosowań jest bardzo zbliżony, bardzo trudno jest jednoznacznie odpowiedzieć na to pytanie. Action Replay posiada dużo więcej funkcji których mi osobiście brakuje w X-Powerze. Mam tu na myśli moduły umożliwiające ciągłą kontrolę wektorów systemowych, zerowanie pamięci po każdym resecie, edytor mapy klawiatury itp. Skazany jestem natomiast na X-Power'a, ponieważ w przeciwieństwie do AR-III nie kłóci się z moim kontrolerem twardego dysku. Polecałbym go jednak tym użytkownikom, ktorzy pragną przede wszystkim "wycinać" obrazki, sample, moduły muzyczne itp. Dla pozostałych polecałbym z kolei AR-III gdyż jego oprogramowanie mimo tego, że zdecydowanie trudniejsze w obsłudze, nie tak estetyczne i nie zawsze stuprocentowo pewne, posiada po prostu więcej możliwości. Regułą musi się jednak niestety stać sprawdzenie poprawności działania w przypadku posiadania jakichś innych urządzeń podłączanych w to samo miejsce co i cartridge np. twardy dysk, fast-RAM, Scanner itp.



... a tak jego działanie na ekranie

SD!

Ogłoszenia drobne

Polski, dyskowy Amiga-Magazyn "QWERTY" = 15.000 zł lub dysk+koperta+znaczek
Krzysztof Latuszek
ul. Ułańska 14 Wrocław

Amiga - Gry, użytki, dema. Katalog: koperta + znaczek
Radosław Twardzik
ul. Przemysłowa 31
22-100 Chełm

C64+magnetofon+stację1541-II, oprogramowanie, literaturę, 3x Cartridge zamienię na C-128+1571
Dominik Latusek
Os. Kosmonautów 23/4
61-642 Poznań

Nawiążę kontakt z posiadaczami Amigi 500. Wymiana programów i doświadczeń.
Krzysztof Leśniewski
ul. Żeromskiego 52a/1
44-119 Gliwice 3

Proponuję wymianę oprogramowania i doświadczeń
Dominik Kulawik
ul. Tuwima 14a
34-500 Zakopane

Kupię książki: J. Ruszczyca "Assembler 6502", B. Frelka "Commodore 64", "Programowanie w kodzie masz."
Piotr Hryciuk
08-201 Niemojki

Sprzedam Public Domain na C-64 oraz literaturę komputerową (koperta+znaczek.
Marcin Lis
ul. Kochanowskiego 14b/20
01-864 Warszawa

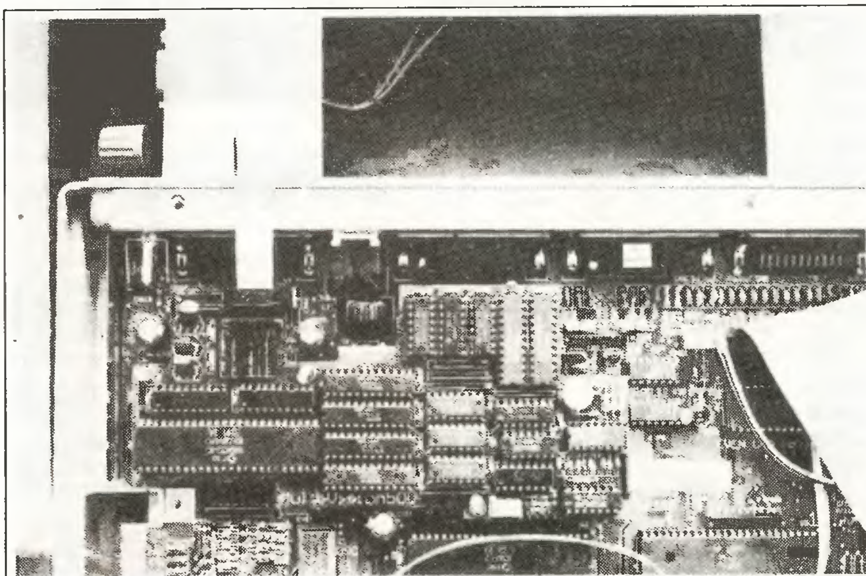
Sprzedam C-64 z magnetofonem, 2 joystickami, cartridgami, programami i literaturą.
Maciej Mołek
ul. Daszyńskiego 7/2
88-100 Inowrocław



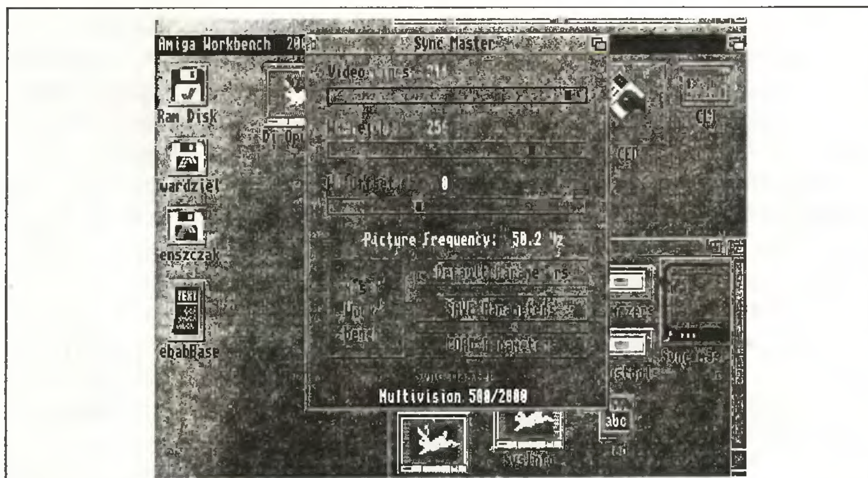
Interlace...? O tak!

Multivision 500 - Flicker-Fixer

Od samego początku swojej historii, Amiga była bardzo kontrowersyjnym komputerem mimo jej niewątpliwych zalet ciągle krytykowana to za to, to za tamto. Aktualnie, gdy znalazła już swoje miejsce na rynku komputerowym i wypowiedzi nawet najzagorzalszych "pecetowców" można traktować z pobłażaniem a na prawie każdy stawiany (najczęściej na skutek braku wiedzy) zarzut można dać jednoznaczną ripostę, ciągle solą w oku Amigowców pozostaje jedno: INTERLACE! Ten całkiem ciekawy skądinąd pomysł (wynalazek) potrafi spędzić sen z oczu niejednego z nas. Kilkudziesięciominutowe wpatrywanie się w "mrugający" ekran potrafi przyprawić o ból głowy i zapalenie spojówek nawet bardzo odpornych osobników. Jak próbowano sobie z tym poradzić na przestrzeni wieków (Amigi)? Pierwszym pomysłem było wprowadzenie na rynek specjalnego monitora o wydłużonym czasie poświaty. Przypominają mi się tu oryginalne monitory firmy IBM zpoczątku lat 80-tych, na których po wyłaczeniu zasilania można było jeszcze przez kilkanaście minut odczytać w ciemności zawartość ostatniego ekranu. No cóż widać pomysł nie nowy, można spróbować jeszcze raz. Spróbowano! Skutki łatwe były do przewidzenia. Obraz w trybie interlace mrugał dalej, co prawda znacznie mniej, ale było to zaletą jedynie w przypadku obrazów nieruchomych. Każdy ruch (animacja) ciągnął za sobą całą chmarę "duchów". Odpowiedź rynku była jednoznaczna. W końcu nie po to się kupuje amigę, żeby oglądać nieruchome obrazki. Potrzeba było czegoś lepszego. Po pewnym czasie znalazło się rozwiązanie. Jakie? Po prostu najzwyklejszy w świecie Flicker-Fixer (ha-ha). Kosztowało to ogromne pieniądze (prawie tyle co Amiga), oprócz tego trzeba było jeszcze kupić specjalny monitor (ten normalny się obraził i do dzisiaj nie chce współpracować z żadnym Flicker-Fixer'em) co dawało w sumie astronomiczną kwotę nie dającą się w żaden sposób wysupłać z kieszeni taty.Tak więc przez długi czas Flicker-Fixer'y (pojawiło się ich w międzyczasie sporo) były tylko ciekawostką dla najbardziej maniakalnych fanów Amigi. Pewnie byłoby tak nadal gdyby nie... "Multivision". Ten pierwszy z fixerów, który kosztuje tyle, że można sobie na niego pozwolić nawet przy niewielkim darze przekonywania (rodziców), nie ustępuje w niczym swoim dwa i trzy razy droższym konkurentom. Zasada działania wszystkich urządzeń tego typu jest taka sama: zapamiętać oba wyświetlane na przemian w trybie interlace półobrazy, a następnie wyświetlić je jednocześnie na ekranie monitora. Niestety wymaga to innej częstotliwości synchronizacji obrazu a standardowy monitor od Amigi potrafi tylko w bardzo wąskim zakresie dostosować się do różnych podawanych z komputera częstotliwości. Byłby to spory zgrzyt, gdyby się okazało, że mamy Flicker-Fixer, a nie mamy go do czego podłączyć! Na szczęście inni konstruktorzy (nie od ulepszania Amigi) też nie próżnowali i wymyślili monitory najpierw dwu, a później wieloczęstotliwościowe te ostatnie nazwano eleganckim słowem "multisync". Monitory te są w stanie automatycznie dostosować się do każdej sensownej częstotliwości synchronizacji, w tym również i do takiej jaka wychodzi z naszego komputera wyposażonego we fli... no właśnie tyle tylko, że najtańsze monitory typu multisync kosztują nie mniej jak 1000,- DM w Niemczech



Multivision 500 na swoim miejscu



i program do niego dołączany

a po przywiezieniu do Polski około 10 milionów złotych. Okropność! Ale... nie zrażajmy się jeszcze. Bo tu wreszczie na coś się przydały komputery typu PC. Ich konstruktorzy po wielu nieprzespanych nocach stworzyli wreszczcie jakąś sensowną kartę graficzną, potrafiącą już prawie tyle co każda najzwyklejsza nawet Amiga (no! może nie całkiem - brakuje jeszcze sprite'ów, blitter'a i trzy tysiące kilku kolorów na ekranie) i nazwali ją VGA. Oczywiście skonstruowano też do niej odpowiedni monitor, który trafem jakimś niemalże idealnie pasuje częstotliwością synchronizacji do naszego FFa (skrót od Fli-Fi). Tak więc kolorowy monitor typu VGA kosztuje mniej więcej tyle co standardowy monitor do Amigi, czyli monitor za monitor. Pozostaje tylko dokupić FF i mamy sprawę "migania" z głowy. Przechodząc teraz konkretnie do naszego "Multivision", to można go kupić zarówno w wersji dla A-500 jak i A-2000. W przypadku "pięćsetki" (również 500plus) instalujemy go w podstawkę która pozostanie na płycie po (umiejętnym!) wyjęciu układu Denise. Sama Denise znajdzie swoje miejsce w podstawce na płytce Fixer'a. Oprócz tego dostajemy jeszcze małe pudełeczko które musimy podłączyć do normalnego wyjścia RGB Amigi. Posiada ono z jednej strony gniazdo do podłączenia monitora VGA (lub multisync), z drugiej gniazdo do podłączenia normalnego monitora RGB, z trzeciej dwa gniazda do podłączenia głośników... No właśnie przecież monitory VGA nie mają wogóle głośników, ani wejścia na fonię. W tym właśnie celu umieszczono w dodatkowym pudełeczku wzmacniacz m.cz. umożliwiający podłączenie dwóch głośników zewnętrznych. Regulacja głośności odbywa się pokrętłem umieszczonym z tej samej strony co gniazdo monitora VGA. Po zainstalowaniu może się okazać, że obraz wcale nie jest taki jak byśmy oczekiwali na przykład "skacze" (tzn. nie trzyma synchronizacji) albo jest "rozmyty" (tzn. nie ma wyraźnych konturów) lub jeszcze inne dodatkowe efekty. Wszystko to nie musi wystąpić, ale może. W pierwszym przypadku sprawa leży w monitorze. O ile monitor posiada (najczęściej z tyłu lub w środku) pokrętło lub potencjometr opisany: "VSYNC" lub coś podobnego, to można korzystając z niego ustalić stabilny obraz (choć czasami może się zdarzyć, że zabraknie zakresu).

W drugim przypadku sprawa leży w Amidze i należy, zgodnie z instrukcją obsługi, po zamontowaniu zestroić każdy egzemplarz fixera ze swoją Amigą. Do urządzenia dołączana jest dyskietka ze specjalnym oprogramowaniem. Przy jego pomocy można np. zwiększyć "częstotliwość ramki" nawet do 100Hz. A także dopasować się do każdego monitora VGA. W naszym warsztacie przetestowaliśmy trzy monitory: monochromatyczny monitor VGA firmy Commodore oraz dwa kolorowe monitory VGA typu "noname" różnej konstrukcji. Staraliśmy się nie korzystać z załączonego oprogramowania. W przypadku monitora Commodore koniecznym okazało się zdjęcie obudowy, odszukanie potencjometru odpowiedzialnego za synchronizację pionową i odpowiednia regulacja. Drugi z testowanych monitorów miał wyprowadzone na zewnątrz właściwe pokrętło regulacyjne. Jednakże pomimo ustawienia tegoż pokrętła w skrajne położenie wymagał (po włączeniu) kilkunastu sekund na dopasowanie się do impulsów podawanych z komputera. Trzeci monitor nie dał się z braku potencjometru regulacyjnego dopasować do sygnału bez użycia metody software'owej. Właściwie to w ten sposób dopasowaliśmy sygnał do monitora a nie odwrotnie. Tak po początkowych perypetiach staliśmy się użytkownikami Flicker-Fixer'a. Muszę przyznać, że daje to dużą satysfakcję. Urządzenie mimo swojej stosunkowo niskiej ceny pracuje bez zarzutu. Jakość obrazu we wszystkich trybach Amigi nie budzi żadnych wątpliwości. Odtwarzanie barw na obydwu kolorowych monitorach - bez zastrzeżeń. Multivision, tak jak wszystkie prawie współczesne FFy nie tylko likwiduje migotanie obrazu w trybie interlace, ale również, poprzez likwidacje (wypełnienie) czarnych prążków międzyliniowych, poprawia znacznie obraz w trybie nolace. Dosyć niespodziewanym efektem ubocznym poprawionej również ostrości konturów, jest uwidocznienie krawędzi wszelkich "płynnych" przejść barwnych oraz niedoskonałości konstrukcji fontów wykorzystywanych przez system do komunikacji z użytkownikiem. Litery stają się "kanciaste". Nie jest to w żadnym przypadku wina Flicker-Fixer'a. Po prostu na standardowych monitorach RGB do Amigi umyka to uwadze ze względu na niedoskonałość (brak ostrości)





tych monitorów. Do wad "Mulitivision 500" należy zaliczyć niezbyt przemyślaną konstrukcję dodatkowego pudełeczka dołączanego do gniazda monitorowego. Stosunkowo najmniejszym błędem wydaje się być konieczność przeprowadzenia kabla (taśmy wieloprzewodowej) łączącego płytkę główną urządzenia ze wspomnianym pudełeczkiem pomiędzy górną i dolną częścią obudowy komputera. W przypadku gdy użytkownik chce po dokonanej instalacji skręcić ponownie obudowę, może napotkać na pewne trudności. Większą wadą (a przede wszystkim łatwiejszą do usunięcia przez konstruktorów) wydaje się być takie rozmieszczenie gniazd głośników, które (w przypadku chęci skorzystania z nich) jeżeli nie uniemożliwia, to bardzo utrudnia podłączenie... zasilacza! Z kolei pokrętło regulacji głośności umieszczone jest tak, że uniemożliwia wykorzystanie gniazda monochromatycznego sygnału wizyjnego. Widać konstruktorzy uznali, że jak stać nas na zakup Amigi, Flicker-Fixer'a oraz monitora co najmniej VGA to z pewnością mamy już i zestaw Hi-Fi do którego podłączymy fonię z komputera. Co do gniazda MONO, to pewnie nikt mądry (według ich zdania) nie będzie korzystał z tego gniazda mając możliwość podłączenia się do dwóch innych (VGA i standard RGB). Opłaca się podłączyć dwa monitory do w/w gniazd chociaż raz po to aby zobaczyć różnicę.

SD!

64

Kupiłem C-64...

i co dalej? (cz.3)

W poprzednich odcinkach naszego cyklu "Kupiłem C64 i co dalej?" zajeliśmy się zupełnymi podstawami, dotyczącymi sposobu podłączenia komputera do domowego telewizora, udzieliliśmy paru rad odnośnie sposobów kopiowania programów itp. W obecnym wydaniu znajdziecie parę rad dotyczących programowania w języku BASIC. Pomijam tutaj dobre rady różnych specjalistów, którzy twierdzą, iż BASIC wypacza pewne prawidłowe podejście do programowania (tutaj zwykle porównuje się BASIC z innymi językami tzw. wyższego rzędu). Faktem jednak pozostaje, że po włączeniu naszego ukochanego Commodore 64 do pracy zgłasza się nam zainstalowany na stałe interpreter języka BASIC, który może być nam przydatny do wielu zastosowań. Nie chciałbym się tu wdawać w tłumaczenie zasady pracy tego języka używając do tego celu jakiejś skomplikowanej terminologii (w każdym bądź razie jeszcze nie teraz). Uważam, że osoby bardziej zainteresowane sięgną w tym celu po odpowiednią literaturę, która zaspokoi ich potrzeby wiedzy w tym zakresie. Jak więc w prosty sposób wykorzystywać wbudowany standardowo interpreter? Jako pewnego rodzaju przybliżenie możemy przyjąć, że BASIC umożliwia obróbkę informacji i jej wydruk na ekranie. Składa się on z kilkudziesięciu komend (tzw. kluczy), które wprowadzone wraz z parametrami wykonują określone operacje. Interpreter j. BASIC zainstalowany w C64 posiada od razu zintegrowany edytor ekranowy, który służy do wprowadzania rozkazów, które są następnie rozpoznawane przez komputer (rozkaz czy nie rozkaz) i w przypadku ich rozpoznania - wykonywane. Do podstawowych komend należy dyrektywa PRINT (ang.drukuj). Jak ona działa? Napiszmy na ekranie (korzystając z klawiatury a nie np.pisaka):

```
PRINT
```

i wciśnijmy klawisz "Return". Co się stało? Trudno powiedzieć, ale komputer wypisał pod spodem "READY." i pojawił się ponownie kursor ("mrugający kwadracik"). Rozkaz został jednak rozpoznany, gdyż w przeciwnym razie na ekranie ukazyłby się komunikat "Syntax Error", który oznacza błąd składni (rozkaz nie rozpoznany). A spróbujmy teraz tak:

```
PRINT 1000
```

Po wciśnięciu "Return" komputer wypisał w następnej linii liczbę 1000. Co to oznacza? Oznacza to, że rozkaz PRINT drukuje na ekranie to, co się za nim znajduje (w tym wypadku - 1000). Działalność PRINT nie ogranicza się tylko do prostego drukowania tego, co za nim umieszczono - zobaczcie przykład:

```
PRINT 2+5
```

Tym razem na ekranie ukazało się nie "2+5" ale gotowy wynik tego dodawania, czyli "7". W ten prosty sposób możemy wykorzystywać komputer jako podręczny kalkulator, który bez problemu jest w stanie przyjąć takie wyrażenia jak np.:

```
PRINT24*(17.23+SIN(12))/23-SQR(12)
```

itp.

I nie musimy się martwić o kolejność działań, ponieważ komputer ustala ją automatycznie. PRINT jest w stanie drukować również teksty:

PRINT "tekst"

Liczby i tekst w instrukcji PRINT mogą być drukowane przy użyciu



pojedynczego PRINT'a, ale w takim wypadku należy użyć pewnego rodzaju połączeń (na zasadzie spójników w j.polskim), którymi są znaki ";" i ",". W uproszczeniu ich działanie można opisać jako "sklejaj" (dla średnika) i "zrób odstęp" (dla przecinka). Proponuję sprawdzić samemu, jak to działa:

```
PRINT 1,2,3,4;5;6
PRINT "ala";"as"
PRINT "ala","as"
PRINT "ala";12,"tekst"
```

Myślę, że podane informacje dotyczące rozkazu PRINT na razie Wam powinny wystarczyć. Zauważyliście na pewno, że takie korzystanie z BASIC ma jedną wadę - komendy wydawane są w trybie bezpośrednim i są przez to jakby "jednorazowego" użytku. Aby temu zaradzić, BASIC pozwala na zapisanie wydawanych mu poleceń w postaci programu, gdzie zostają one ułożone w kolejności ich wykonywania. Aby zapisać jakieś polecenie w postaci linii programu, wystarczy przyporządkować mu odpowiedni numer, i tak np.:

```
10 PRINT 2+2
```

po wciśnięciu "Return" staje się automatycznie linią programu o numerze 10. Uruchomić taki program można przy użyciu rozkazu RUN. Wpisanie innego polecenia z numerem linii równym 10 powoduje skasowanie starej linii o tym samym numerze i wpisanie aktualnej na jej miejsce. Aby skasować wybraną linię wystarczy wpisać tylko jej numer i wcisnąć "Return". Jeżeli chcielibyśmy zobaczyć efekt naszej pracy i kazać sobie wydrukować na ekranie całość wpisanego programu, wystarczy użyć do tego celu rozkazu LIST. W dzisiejszym odcinku pozostała do omówienia jeszcze jedna ważna sprawa - tzw. zmienne. O co tu chodzi? A pamiętacie ze szkoły jakikolwiek wzór, np. na pole powierzchni prostokąta? Długości boków zastąpiono w nim literami A i B po to, by można było pod nie podstawić dowolne wartości. W podobny sposób można to robić w jęzuku BASIC:

```
10 A=10
20 B=20
30 C=A*B
40 PRINT "Pole prostokata:";C
```

Jak zapewne się już domyślacie, w lini 10 zdefiniowano zmienną A jako jeden z boków prostokąta, w 20 zmienną B jako drugi bok. W lini 30 natomiast zdefiniowaniu ulega zmienna C, której przypisywana jest wartość z wyrażenia "A razy B". Podczas definiowania nazwy zmiennej należy pamiętać o tym, że musi się ona zaczynać od litery i nie może być jednocześnie którymś z rozkazów BASIC lub jedną z zarezerwowanych zmiennych systemowych, o których wspomnimy w następnym odcinku.

W podobny sposób można definiować zmienne tekstowe, które w programie wyróżniają się przede wszystkim tym, że ich nazwy są zakończone obowiązkowo znakiem "$" (np. a$, tekst$ itp.). Ich definiowanie przebiega następująco:

```
10 A$="tekst 1"
20 B$="tekst 2"
30 C$=A$+B$
```

Jak widać na przykładzie, zmienne tekstowe mogą być łączone przy pomocy znaku "+". Nietrudno zauważyć, że nadanie jakiejś zmiennej nowej wartości oznacza zmianę tekstu programu. Aby tego uniknąć wprowadzimy sobie instrukcję INPUT (ang.włóż), która pozwala na nadanie zmiennym odpowiednich wartości bez konieczności modyfikacji programu. Może prześledźmy przykład:

```
10 INPUT "Podaj dlugosc boku A:";A
20 INPUT "Podaj dlugosc boku B:";B
30 C=A*B
40 PRINT "Pole prostokata:";C
```

Na koniec tego odcinka jeszcze dwie informacje: zamiast pisać całe słowo PRINT, wystarczy użyć zastępczo znaku "?", który dla wygody użytkownika oznacza to samo; i druga informacja - do skasowania programu w języku BASIC służy polecenie NEW, które pozwala nam w ten sposób na rozpoczęcie pisania programu od nowa.

Paweł "Polonus" Sołtysiński

GYROS!









Mapa pamięci Amigi

(c.d.)

020 DSKPTH Z * + A
022 DSKPTL Z * + A
Wskaźnik danych dla dysku. Do tych rejestrów należy wpisać adres pod którym znajdują się dane dla operacji dyskowych (zapis/odczyt), zanim odpowiedni kanał DMA zostanie zainicjowany. Należy pamiętać o tym, że wpisany adres powinien wskazywać pamięć typu CHIP i być wartością parzystą (najmłodszy bit skasowany).

024 DSKLEN Z * P
Długość operacji dyskowej. Zawiera ilość słów do przesłania podczas operacji dyskowej, oraz dwa bity kontroli. Poszczególne bity oznaczają:
bit 15 (DMAEN) Uaktywnia kanał DMA dla dysku.
bit 14 (WRITE) Odpowiedzialny za kierunek transmisji. Ustawiony oznacza zapis czyli wyprowadzanie z pamięci na dysk, skasowany - przeciwnie.
bity 13-0 (LENGTH) Zawierają ilość słów do przesłania.

026 DSKDAT Z & * P
Rejestr zapisu danych dla operacji dyskowych. Rejestr ten jest zapisywalny przez wewnętrzne kanały DMA i użytkownik nie posiada do niego dostępu nawet z poziomu języka maszynowego.

028 REFPTR Z & * A
Rejestr odświeżania wskaźników. Używany przez kanały DMA, nigdy nie powinien być zmieniany przez mikroprocesor.

02A VPOSW Z * A
Zapis najbardziej znaczącego bitu pozycji pionowej rastra. Używany głównie w obsłudze tzw. przerwań rastrowych czyli takich, które mają zostać wykonane, gdy wyświetlanie ekranu dojdzie do zadanej pozycji (przerwanie opisane bitem 4 rejestru 01C). Znaczenie poszczególnych bitów jest takie same jak w rejestrze VPOSR.

02C VHPOSW Z * A
Zapis pozycji poziomej i pionowej rastra. Używany w parze z VPOSW. Znaczenie poszczególnych bitów jest takie same jak dla rejestru VHPOSR.

02E COPCON Z * A
Rejestr kontroli Copper'a. W rejestrze tym używany jest tylko jeden bit zwany CDANG (bit 1). Jego ustawienie umożliwia Copper'owi dostęp, i zapis pewnej grupy rejestrów odpowiedzialnych za pracę Blitter'a. Może teraz parę słów o samym Copperze. Co mieści się pod tą nazwą? Otóż nic innego jak wyspecjalizowany układ służący przede wszystkim do opisu i kontroli generowanego obrazu. Posiada bezpośredni dostęp do pamięci, przez co może pracować niezależnie od procesora głównego. Za jego pomocą możemy obsługiwać sprite'y, zmieniać aktualne kolory na ekranie, inicjować kanały Audio, oraz jak już wspomniałem wyżej sterować pracą Blitera. Sprawne programowanie Copper'a czyli sporządzanie tyw. Copper-List'u jest bardzo istotne i zostanie szczegółowo opisane przy okazji innych rejestrów związanych z tym układem.

030 SERDAT Z * P
032 SERPER Z * P
034 POTGO Z * P
036 JOYTEST Z * D
038 STREQU S & * D
03A STRVBL S & * D
03C STRHOR S & * DP
03E STRLONG S & * D

Począwszy od pozycji 040 zaczynają się rejestry obsługujące Blitter'a. Jest to drugi, obok Copper'a koprocesor Amigi, mogący pracować w dwóch trybach. Pierwszym z nich jest tzw. "block mode" w którym to, potrafi z ogromną prędkością wykonywać dowolne operacje logiczne na blokach pamięci, opisanymi trzema rejestrami źródłowymi (ang: source), aby ostatecznie uzyskany wynik zapisać w obszarze przeznaczenia (ang: destination). W szczególnym przypadku, Blitter może po prostu kopiować uprzednio zdefiniowane okna pamięci z miejsca źródła do miejsca przeznaczenia. Ponadto istnieje możliwość wypełniania zamkniętych obszarów oraz przesuwania o dowolną z zakresu 1-15 liczbę bitów, bloku podczas wyżej wspomnianych operacji logicznych. Tę ostatnią cechę wykorzystuje się przy pisaniu procedur scroll'i. Drugim trybem pracy Blitter'a jest tzw. "Line Mode". Umożliwia on kreślenie linii z prędkością bliską milion punktów na sekundę, przy czym kreślona linia nie musi być ciągła, ale może zostać dowolnie zdefiniowana przez programistę. Blitter posiada kanały bezpośredniego dostępu do pamięci (DMA) i dzięki temu może pracować jednocześnie z procesorem głównym. Posiada on jednak jedno ograniczenie: mianowicie może adresować jedynie pamięć typu CHIP. Dokładny opis rejestrów tego układu rozpocznę w przyszłym numerze.

Krzysztof Kobus



Amos The Creator

Opis systemu - cześć I

Jeszcze nie tak dawno jedynym interpreterem Basic'a na Amigę był beznadziejnej jakości Microsoft Basic. Niestety dostarczany na dysku Extras program był bardzo podłej jakości. Edytor był wręcz tragiczny, przykłady bamalne a prędkość całkiem przeciętna. Microsoft Basic nie był programem ma miarę Amigi. Wielką innowacją było pojawienie się GFA Basic'a - standardu sprawdzonego na ST. Szybki interpreter, bogactwo funkcji oraz dostępny kompilator uczyniły ten program popularnym językiem programowania. Wiele profesjonalnych programów zostało napisanych właśnie w GFA Basic'u. GFA był naprawdę bardzo dobrym programem. Wydawać się mogło że jeśli chodzi o Basic nic lepszego nie można już wymyśleć. A jednak! AMOS Basic to nowy rewelacyjny interpreter/kompilator tego powszechnie znanego języka programowania. AMOS Basic to prawdziwy krok na przód w dziedzinie programów narzędziowych na Amigę. Programiści AMOS'a zadbali o to aby wraz z nim dostarczyć całe bogactwo programów pomocniczych (napisanych oczywiście w AMOS Basic-'u). Uwalnia to ewentualnego programistę od zbędnych kłopotów z wymianą danych między różnymi narzędziami. Jedynymi dodatkowymi narzędziami może być Deluxe Paint oraz Sound Tracker (ew. inny program muzyczny). AMOS Basic oferuje bogactwo funkcji. Oprócz łatwego dostępu do setek funkcji bibliotek systemowych zawiera ponad tysiąc własnych procedur. Program jest napisany bardzo starannie - rzadko zawiesza się i nie wykazuje wad w funkcjonowaniu.

AMOS - na pierwszy rzut oka

Na system składają się dwa dyski. Na pierwszym znajduje się sam AMOS, przykłady do manuala oraz liczne programy użytkowe. Drugi zawiera cztery gry napisane w AMOS'ie oraz banki ze sprite'ami, muzyką i grafiką. Do pracy niezbędny jest jedynie pierwszy dysk.

Wymagania sprzętowe

AMOS dość sprawnie pracuje z 512 KB RAM jednak aby wykożystać w pełni możliwości programu wskazane byłoby mieć 1MB. Ponieważ program nie jest nakładkowany zatem dodatkowe stacje dysków nie są niezbędne.

Pierwsze kroki ...

Program ładuje się automatycznie po zabootowaniu pierwszego dysku. Efektownie wjeżdża na ekran winieta tytułowa. Górna część ekranu zajęta jest przez banderolę, z której można wybierać opcje oraz linię statusową. Dolną wypełnia pełnoekranowy edytor tekstu porównywalny z Cygnus Pro.

Menu

Istnieje pięć zestawów menu. Pierwszy jest zawsze widoczny i jest wybierany przez kliknięcie na odpowiednim polu ewentualnie wciśnięcie klawisza F1-F10. Przez wciśnięcie klawiszy SHIFT (równoważny prawemu klawiszowi myszy), CTRL lub ALT otrzymujemy inne menu.

Krzysztof Moroń

Amiga & C64 Copy-Party

W dniach 28/29 marca w Warszawie odbyło się Copy-Party, zorganizowane przez grupy: Action Direct, G-Force, Katharsis. A oto relacja naszych korespondentów:
Po ośmiogodzinnej podróży przebytej bez większych przygód w pociągu Polskich Kolei Państwowych, około godziny 8 wysiedliśmy na dworcu w Warszawie. Nie zwlekając, skierowaliśmy w stronę liceum i za kwadrans byliśmy już na miejscu. Tutaj okazało się, że nie byliśmy pierwsi. W sali gimnastycznej, na uprzednio przygotowanych stołach rozłożyli się już przybyli przed nami amigowcy i komodorowcy. Znaleźliśmy następnie wolne miejsce, rozpakowaliśmy bagaże, i usiedliśmy zmęczeni podróżą. Nasza radość spowodowana szczęśliwym dotarciem na od dawna oczekiwane party nie trwała długo - w pewnej chwili przyszedł jeden z organizatorów i grzecznie, aczkolwiek stanowczo wyprosil towarzystwo z sali gimnastycznej na hall tłumacząc: "Będziemy podłączać". Na szczęście sprzęt i bagaże mogły zostać w sali. Nie pozostało więc nic innego jak przyjąć prośbę organizatorów i nie przeszkadzać im, w i tak ciężkiej pracy. Posłusznie wyszliśmy do hallu, gdzie usiedliśmy czekając na dalszy rozwój wypadków. Jak się później okazało nie trwało to długo. Tym razem przemówienie innego już organizatora było bardziej apodyktyczne. Rozkaz brzmiał: "Opuścić szkołę i przyjść o 10-tej". Na nieśmiałe głosy protestu usłyszeliśmy od człowieka noszącego wizytówkę ECL/KTS słowa, których cytować nie będę, wspomnę tylko, że były wypowiedziane łamaną angielszczyzną, a dociekliwych informuję, że w ostatnich czasach często można przeczytać je na murach wszystkich miast. Cóż, do godziny 10-tej zostało 90 minut, na dworze zimno, więc wróciliśmy na dworzec, gdzie w miejscowym barze spędziliśmy wolny czas. Punktualnie o wyznaczonej godzinie z powrotem zjawiliśmy się pod szkołą. Tym razem nie zawiedliśmy się. Zaczęło się punktualnie! Wpłaciliśmy do kasy 40 tys. złotych, i... znów zdziwienie. Nie dostaliśmy wizytówek, ani biletów, natomiast każdy został ostemplowany, co jak się później okazało było żelaznym glejtem i zapewniało swobodny wstęp na imprezę, po uprzednim wyjściu na przykład do miasta. W przypadku jednak, gdy komuś podczas tych dwóch dni przyszło do głowy umyć się znalazłby się w bardzo kłopotliwej sytuacji: pieczątki znikały. Na szczęście organizatorzy okazywali w tej kwestii pełne zrozumienie. Tak wyglądały nasze początkowe perypetie. A co z samym party? Tutaj wiele niespodzianek. Przede wszystkim niewątpliwą nobilitacją imprezy było pojawienie się znanej, zachodniej grupy SANITY. Przywieźli oni ze sobą swoje najno-

ciąg dalszy na stronie 16

64

Assembler na C-64

(odc. 3)

Na początek kolejnego odcinka naszego kursu, jak zwykle kilka wyjaśnień odnośnie króciutkich listingów, które znalazły się w poprzednim. Również i tym razem były to dwa programy realizujące to samo zadanie, ale napisane w różnych językach. W naszym przypadku był to BASIC oraz język asemblera. Zadaniem każdego z nich było wypełnić cały ekran znakiem "@" a następnie go wyczyścić. Jeżeli spróbowaliśmy je wpisać (i uruchomić) to mieliśmy możliwość po raz kolejny ujrzeć potęgę (szybkość) kodu maszynowego. Należy tu zwrócić uwagę na fakt, że program maszynowy nie został napisany w taki sposób aby uzyskać maksymalną szybkość lecz przejrzystość. W praktyce istnieje możliwość osiągnięcia jeszcze znacznie większej szybkości wykonywania. W jaki sposób uzyskaliśmy zamierzony cel? Po prostu odwołaliśmy się bezpośrednio do pamięci ekranu... ojej, zapomniałem, że nie każdy musi od razu wiedzieć co to jest pamięć ekranu! Spróbuję to jakoś wytłumaczyć. Co to jest obszar adresowy to powinniśmy już wiedzieć (było w poprzednim odcinku). W całym obszarze adresowym (co również widać na schemacie z poprzedniego odcinka), rozciąga się pamięć RAM (64K). Spośród tych 64 kilobajtów pamięci wyodrębnijmy teraz jeden kilobajt rozpoczynający się od adresu $0400 (1024). To właśnie tam znajduje się po włączeniu komputera w/w pamięć ekranu. Jest to po prostu wybrane z całej pamięci 1000 bajtów przeznaczonych do przechowywania zawartości ekranu. Każdemu z tych bajtów odpowiada jeden znak na ekranie. Jednym zdaniem wszystko co umieścimy w tym obszarze pamięci będzie natychmiast widoczne na ekranie. Spróbujmy! Napiszmy np.:

```
PRINT CHR$(147)
POKE 1523,1:POKE
1524,2:POKE 1525,3
```

Te rozkazy (w BASIC'u oczywiście) umieszczają kolejno pod zadanymi adresami (w naszym przypadku: 1523, 1524 i 1525), wartości 1, 2 i 3. Powinniśmy natychmiast zauważyć efekt tego umieszczenia w postaci pojawienia się na środku ekranu liter "ABC". Jeżeli mamy bardzo starą wersję komputera (z lat 1982 do 1984) to może się zdarzyć, że po wpisaniu POKE'ów nic się nie pojawi. Spowodowane to jest niewielkim błędem w systemie operacyjnym (KERNAL) który został poprawiony dopiero pod koniec roku 1984. Wszyscy Czytelnicy, którzy mają takie właśnie komputery, muszą wpisać jeszcze jedną linijkę:

```
FOR I=55296 TO
56319:POKE I,1:NEXT I
```

Najlepiej wpisać to przed POKE'ami. Efektem wpisania tej drugiej linijki jest (we wszystkich wersjach C-64) zmiana koloru wszystkich liter na biały. Oczywiście mam nadzieję, że nie muszę dodawać, że wszystkie linie należy wpisać w tzw. trybie bezpośrednim BASIC'a tzn. bez numeru oraz wcisnąć RETURN po każdej z nich. Wracając do naszej pamięci ekranu. Wiemy już, że poprzez umieszczanie w komórkach pamięci ekranu odpowiednich wartości, możemy uzyskać efekt pojawiania się różnych znaków na ekranie. Pamiętajmy o tym, że spacja to też jest znak, mimo tego iż go nie widać. Co to oznacza? Oznacza tyle, że ekran "pusty" lub "wyczyszczony" wcale nie jest pusty lecz jest w istocie wypełniony właśnie znakami spacji. Czy komuś już coś świta? Tak! Zajmiemy się teraz właśnie "czyszczeniem" ekranu. Jak się do tego zabrac? W trybie bezpośrednim wciskamy po prostu klawisze SHIFT+CLR/HOME. W BASIC'u zazwyczaj piszemy PRINT CHR$(147) lub PRINT "{CLR}". TO dziwne wyrażenie w nawiasach klamrowych ({}) oznacza wciśnięcie klawisza SHIFT i CLR/HOME. Wszystkie te metody sprowadzają się w końcowym efekcie do WYPEŁNIENIA ekranu (a dokładniej: pamięci ekranu) znakiem spacji któremu odpowiada bajt o wartości $20 (32). Samo wypełnienie wykonują za nas odpowiednie procedury maszynowe systemu operacyjnego, umieszczone w pamięci ROM. My jednak wiemy już prawie wszystko o pamięci ekranu, a przede wszystkim to pod jakim adresem się ona zawsze po włączeniu (lub wykonaniu RESET'u) znajduje i możemy zrobić to sami. Proponuję teraz wziąć do ręki ponownie numer 2/3 KEBAB'a i przyjrzeć się jeszcze raz listingom nr 4 i 5, umieszczonym na ostatniej stronie. Jeżeli wpiszemy tylko linie 40 do 60 z listingu 4, to otrzymamy gotową procedurę czyszczenia ekranu. Spróbujmy! Oto ta sama procedura tym razem w jednej linijce:

```
10 FOR X=1024 TO
2023:POKE X,32:NEXT X
RUN
```

Działa! Tylko czemu tak wolno? No cóż, to jest tylko BASIC. Zróbmy zatem to samo w MC (wiemy wszyscy co ten skrót oznacza) Właści-

wie mamy już to zrobione. Wystarczy w listingu nr 5 z poprzedniego odcinka w miejsce

```
A5012  EOR  #$20
```

wpisać

```
A5012  RTS
```

i nic więcej. Procedura gotowa. Jeżeli teraz ją uruchomimy (najlepiej G 5000 z debuggera, lub SYS 5*4096 z BASIC'a), to na pewno już nie będziemy narzekać na szybkość. Ktoś spostrzegawczy mógłby się jednak zapytać: Po co mi to wszystko? Przecież przed chwilą przeczytałem, że w ROM'ie są gotowe procedury służące do tych samych celów. Tak, to prawda. Można ten sam efekt uzyskać znacznie prościej. Np.:

```
A5000  LDA  #$93
A5002  JSR  $FFD2
A5005  RTS
```

albo jeszcze krócej:

```
A5000  JSR  $E544
A5003  RTS
```

Oba te przykłady odwołują się do gotowych procedur systemu operacyjnego. Występujący w nich rozkaz JSR oznacza (J)ump to (S)ub(R)outine a po polsku "skok do podprogramu". Działa on na podobnej zasadzie co BASIC'owy GOSUB. To znaczy, że procesor gdy napotka podczas wykonywania programu taki rozkaz, musi w jakiś spoób (w późniejszych odcinkach wytłumaczymy w jaki) zapamiętać gdzie go napotkał, aby móc po wykonaniu podprogramu i natknięciu się na komendę RTS, powrócić do programu głównego. Dlaczego wobec tego uczymy się takich procedur skoro są one już na stałe w ROM'ie? Dlatego, że może zdarzyć się w przyszłości, że będziemy pisać programy nie korzystające z systemu operacyjnego a oprócz tego w ten sposób uczymy się dużo więcej na temat samej konstrukcji komputera i jego działania. A teraz gdy już wiemy jak wyczyścić ekran to możemy zabrać się do drukowania jakichś napisów na tym ekranie. Znowu proponuję zacząć od bezpośredniego odwołania się do pamięci ekranu. Na początek zmienimy kolory tła i ramki na jakieś sensowne.

```
A5000  LDA  #$00
A5002  STA  $D020
A5005  STA  $D021
```

Teraz wyczyścimy ekran,

```
A5007  TAX
A5008  LDA  #$20
A500A  STA  $0400,X
A500D  STA  $0500,X
A5010  STA  $0600,X
A5013  STA  $0700,X
A5016  INX
A5017  BNE  $500A
```

zmienimy kolor znaków na białe,

```
A5019  LDA  #$01
A501B  STA  $D800,X
A501E  STA  $D900,X
A5021  STA  $DA00,X
A5024  STA  $DB00,X
A5027  INX
A5028  BNE  $501B
```

i umieścimy biały napis

```
A502A  LDA  #$0B
A502C  STA  $0400
A502F  LDA  #$05
A5031  STA  $0401
A5034  LDA  #$02
A5036  STA  $0402
A5039  STA  $0404
A503C  LDA  #$01
A503E  STA  $0403
A5041  RTS
```

Zarówno podczas czyszczenia ekranu jak i zmiany kolorów liter, używaliśmy nieznanego nam jeszcze wyrażenia "STA $????,X". Wiemy dobrze co to jest "STA $????" (???? - oznacza dowolną czterocyfrową liczbę szesnastkową). Jest to umieszczenie kopii zawartości akumulatora pod adresem ????. Wiemy również że oprócz akumulatora mamy także inne rejestry procesora: X - rejestr indeksowy X, Y - rejestr indeksowy Y. Pewnie się już domyślamy, że ten dziwaczny ",X" w naszych przykładach musi mieć coś wspólnego ze wspomnianym rejestrem (indeksowym) X. Spróbujmy wyjaśnić co oznacza określenie "indeksowy". Napisanie "STA $D020" oznacza wykorzystanie tzw. adresowania absolutnego (ang.: absolute addressing), czyli odwołanie się bezpośrednio do konkretnego adresu (u nas $D020). Napisanie natomiast "STA $D020,X", oznacza użycie adresowania absolutnego z indeksem (ang.: absolute addressing with index). Oznacza to, że do adresu absolutnego ($D020) zostanie dodana wartość indeksu (w tym przypadku wartość znajdująca się w rejestrze X) i dopiero suma tych dwóch składników da nam adres pod którym zostanie umieszczona kopia zawartości akumulatora. No tak, ale skąd wiemy jaka wartość jest aktualnie w rejestrze X? Musimy sobie w nim umieścić odpowiednią wartość sami. Najprościej użyć do tego komendy "LDX". Np.:

```
LDX  #$01
LDA  #$00
STA  $D020
STA  $D020,X
RTS
```

Komenda "LDX #" działa tak samo jak "LDA #" tyle tylko, że odnosi się nie do akumulatora a do rejestru X. W naszym przykładzie "ładujemy X jedynką" (LDX #$01), następnie akumulator zerem, po czym umieszczamy to zero pod adresem $D020 oraz pod adresem $D020+$01 (bo w X mamy $01) czyli razem $D021 no i na końcu jak zwykle wracamy do punktu wyjścia komendą RTS. Do czego nam takie rzeczy potrzebne? W ostatnim przykładzie jest to czysta "sztuka dla sztuki" niemniej jednak już w procedurze czyszczenia ekranu sprawa wygląda zupełnie inaczej. Owszem można by napisać tą procedurę np. w ten sposób:

```
LDA  #$00
STA  $0400
STA  $0401
STA  $0402
STA  $0403
...
STA  $07FF
```

Tylko komu by się chciało pisac ponad tysiąc linii programu skoro można się zmieścić w dziesięciu. W naszym przykładzie (tym drukującym napis na ekranie) wykorzystaliśmy inną możliwość umieszczenia odpowiedniej wartości w rejestrze X. Była to komenda "TAX" tym razem powinno nam się to wyrażenie kojarzyć z angielskim zwrotem "(T)ransfer (A)ccumulator to (X)". Po polsku "umieść kopię zawartości akumulatora w rejestrze X". Ponieważ wcześniej umieściliśmy w akumulatorze wartość zero (LDA #$00) i w międzyczasie nie

zmienialiśmy jej, to po wykonaniu TAX będziemy mieli w X również zero. Następnie załadowaliśmy akumulator liczbą $20 (w kodzie ASCII oznacza to znak spacji) i umieściliśmy tą wartość kolejno pod adresami: $0400+$00 (bo w X mamy $00), $0500+$00, $0600+$00 oraz $0700+$00. Po umieszczeniu spacji (a właściwie liczby jej odpowiadającej) pod w/w adresami pamięci ekranu procesor wykona komendę INX (skrót od (IN)crement (X) czyli zwiększ zawartość rejestru X o 1). Oznacza to, że po jej wykonaniu będziemy mieli w rejestrze X wartość nie $00 a o jeden większą ($01). Następna komenda to BNE $500A. BNE to kolejny skrót od zwrotu "(B)ranch if (N)ot (E)qual (zero)". Oznacza to po polsku: Skocz pod podany (w operandzie) adres jeżeli wynik ostatniej operacji nie jest równy (zero). Dlaczego "zero" napisałem w nawiasie wyjaśnimy w następnych odcinkach. A wracając do przykładu, wynik naszej ostatniej operacji nie był równy zero bo ostatnią operacją było zwiększenie wartości w rejestrze X z $00 do $01 (INX). W związku z tym procesor po odczytaniu komendy "BNE" skoczy (oczywiście nie dosłownie) pod adres $500A czyli do komendy STA $0400,X. Lecz tym razem w rejestrze X jest wartość $01 co oznacza, że ciągle nie zmieniona w akumulatorze wartość $20 zostanie umieszczona pod adresem $0400+$01 następnie $0500+$01 i tak dalej. Następnie kolejne wykonanie rozkazu INX zwiększy o jeden wartość w X (do $02) a kolejne BNE wróci nas znowu na początek pętli. A co będzie gdy w rejestrze X będzie już wartość $FF? Na tym właśnie polega cały trick. Ponieważ wszystkie rejestry (A, X, i Y) są tzw. ośmiobitowe (czyli mogą zawierać liczby z przedziału od $00 do $FF (%00000000 do %11111111) to w przypadku gdy w rejestrze takim mamy wartość maksymalną ($FF) a mimo to wykonamy zwiększenie o jeden (np. INX) to rejestr podobnie jak licznik kilometrów w samochodzie lub licznik taśmy w magnetofonie "przeskoczy" na zero. Tak więc po 256-ciu wykonaniach naszej przykładowej pętli programu razem z "INX" w rejestrze X będziemy mieli ponownie zero i tym razem procesor wykonując komendę "BNE" stwierdzi, że wynikiem ostatniej operacji było zero (bo rejestr X "przeskoczył") i nie wykona już skoku pod adres $500A. Analogicznie rzecz się przedstawia w przypadku kolejnej procedury w przykładzie. Z tą tylko różnicą, że umieszczamy wartość $01 (odpowiednik koloru białego) w obszarze pamięci odpowiedzialnym za tzw. atrybuty znaków. Po angielsku obszar ten nazywa się po prostu "Color RAM" czyli pamięć kolorów. Każdemu znakowi na ekranie (czyli w obszarze $0400 do $07EF) odpowiada jego atrybut w obszarze pamięci kolorów czyli $D800 do $DBEF. Zainteresowanym polecam poeksperymentowanie z różnymi kolorami tj. różnymi wartościami z przedziału $01 do $0F umieszczanymi w linii:

```
A5019 LDA #$0?
```

naszego programu.

SD!

64

Słownik skrótów

oraz terminów "demologicznych"

Prawie od samego poczatku istnienia C64 użytkownicy pisali programy demonstracyjne. Z biegiem czasu stawały się one coraz bardziej wymyślne i zawierały coraz bardziej skomplikowane efekty, które przecież trzeba było jakoś nazwać.I tak powstała cała masa terminów i skrótów określajacych różne efekty w programach demonstracyjnych. W tym cyklu artykułów będziemy starali się je Wam przybliżyć. Dzisiaj zaczniemy od skrótów i nazw opisujących wszelkiego rodzaju animacje.

Od samego początku próbowano poruszać literami po ekranie, ale inaczej niż pozwalały na to rejestry przesuwu i organizacja pamięci ekranu. Ruchy tych literek rożniły się zarówno nazwą jak i stopniem trudności programu. Oto one:

D.Y.C.P. - different Y char position - Jest to płynacy tekst (z ang.scroll), w którym każda literka może mieć inną, płynnie zmienianą pozycję pionową. Przeważnie do określania pozycji znaku używana jest funkcja sinus.

D.X.C.P. - different X char position Tak samo jak w poprzednim przypadku, ale tym razem tekst płynie do góry, a płynnie zmieniana jest pozycja X każdej z literek.

D.X.Y.C.P. - different X Y char position - Połączenie obydwu poprzednich efektów. W praktyce oznacza to, że każda litera może się płynnie poruszać w każdym kierunku.

Lecz programisci poszli dalej i na-

stępnym krokiem było poruszanie każdą pionową kolumną (szerokości 1 punktu) danej literki i tak powstal:

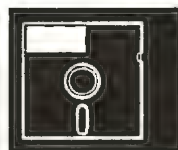
D.Y.P.P. - different Y pixel position - Jak zapewne wszystkim wiadomo, rejestr systemowy $d016 umożliwia poruszanie ekranem o max.8 punktów. Można za jego pomocą "powyginać" obrazek właśnie o max.8 punktów. Po pewnym jednak czasie odkryto sposob na wyginanie ekranu o wiecej niż 8 punktów (w granicach danego znaku).
Efekt ten nazwano **TECH-TECH.** - Jak wiadomo, bezpośrednie, płynne poruszanie ekraniem w HI-RES jest niemożliwe ze względu na szybkość procesora, jednak z biegiem czasu znaleziono zastosowanie pewnych błędów w VIC'u i udało się zmusić go do poruszania ekranem.
Trik poruszajacy ekran na boki nosi nazwę **V.S.P.**, natomias trik poruszajacy ekran w dół nosi nazwe **F.L.D.**
I to juz wszystkie terminy z zakresu animacji znakowo-graficznej. Za miesiąc pomówimy o trikach z użyciem rejestru $d011.

Krzysztof "BRUSH" Dabrowski

Tape - Burger

taśmowy superkopier dla C-64

Jest oczywistym fakt, że firmy wydające oprogramowanie na ogół starają się ze zrozumiałych względów swoje produkty zabezpieczać przed ingerencją niepowołanej do tego osoby. Jako ingerencję taką należy również rozumieć próbę wykonania nielegalnej kopii programu. O ile przyczyny zabezpieczania programów w przypadku tych firm wydają się być jasne, to jak wytłumaczyć fakt używania przez naszych polskich handlarzy (czytaj: złodziei) tego typu technik? Zabezpieczanie nielegalnych kopii? Po co? Oczywiście, z chęci zysku (w tym przypadku płynącego z kradzieży). Mechanizm jest prosty - klient przychodzi, kupuje, w domu próbuje skopiować nowe gry np. koledze - i nic! Istota tkwi w tym, że kupione u handlarza (jak czytać naprawdę podano wyżej) programy w przypadku kupienia ich na gotowej nagranej taśmie, są najczęściej nagrane w ten sposób, że można je wgrać do pamięci za pomocą dowolnego programu przyspieszającego (tzw. turbo), lecz nagrać ich ponownie w działającej postaci nie sposób. Związane jest to ze szczególnym sposobem zapisu danych w spreparowanym zapisie "turbo". Polega to na tym, że zabezpieczona wersja programu zawiera w nagłówku (tytule) nie tylko znaki tworzące nazwę ale również fragment dalszej części programu. W ten sposób wgrany program pracuje bez zarzutu, jednakże podczas próby jego nagrania, program "turbo" zapisuje nagłówek w sposób standardowy, tzn. bez owych dodatkowych danych, które znajdowały się w nagłówkach programów nagranych na zakupionej kasecie. Aby umożliwić naszym Czytelnikom "poradzenie" sobie z tym problemem, powstał program Tape-Burger, który umożliwia kopiowanie tak zabezpieczonych zbiorów (i wszystkich innych też, pod warunkiem, że zapisano je w standardzie TURBO). Program należy wpisać do pamięci komputera używając do tego celu dowolnego monitora (pomijamy wtedy dane zapisane w nawiasach) lub przy użyciu Korektora a następnie nagrać na taśmę lub dysk w/g lokacji podanej przy wydruku.



A teraz trochę o obsłudze, która jest bardzo prosta. W programie zrezygnowano z dodatkowych rozkazów BASIC, które są standardowo używane w innych programach tego typu. Po uruchomieniu program przedstawia się ("Tape-Burger The Ultimate Copier...") i od razu prosi o włożenie kasety z programem przeznaczonym do kopiowania ("press play on tape"). Po uruchomieniu magnetofonu program odczytuje z taśmy nagłówek i kilka informacji o kopiowanym zbiorze (lokacja w pamięci, typ zbioru). Po naciśnięciu spacji ładowanie jest kontynuowane. O ile operacja ta przebiegła bezbłędnie, drukowane jest na ekranie menu, gdzie możemy wybrać jedną z trzech możliwości: załadowanie następnego programu, uruchomienie programu znajdującego się w pamięci oraz jego skopiowanie. Wyboru opcji dokonuje się przez wciśnięcie klawiszy cyfrowych. Miłej pracy z Tape-Burger'em życzy:

Paweł "Polonus" Sołtysiński

Listing nr 1 na stronie 28



Posłuchaj Amigi...

mówiącej ludzkim (i nie tylko) głosem

Wraz z powstaniem pierwszych komputerów narodził się problem łatwej komunikacji między człowiekiem a maszyną. Od tamtego czasu całe sztaby ludzi pracują nad tym, aby użytkownik mógł w miarę naturalny dla niego sposób wprowadzać dane, oraz aby otrzymane wyniki, wyprowadzone były przez maszynę w jak najbardziej przystępnej formie. Niewątpliwie jednym z rozwiązań spełniających powyższe warunki, szczególnie jeśli chodzi o wyprowadzanie wyników, jest synteza mowy. Napisanie programu realizującego syntezę mowy jest stosunkowo proste, tym bardziej, że twórcy systemu operacyjnego Amigi opracowali zespół procedur może nie rozwiązujących tego problemu do końca, lecz na pewno maksymalnie go upraszczających. Czytelników bliżej zainteresowanych tym tematem proszę o zapoznanie się z programem zamieszczonym na końcu Kebaba, oraz krótkim komentarzem.

Do obsługi syntezy mowy służy biblioteka "translator.library", oraz urządzenie "narrator.device". Jakie jest ich zadanie? Otóż wspomniana biblioteka zamienia zwykły tekst (no może nie tak zupełnie zwykły, ale ten który ma zostać powiedziany przez komputer) na ciąg tzw. fonemów, czyli podstawowych elementów dźwięku uwzględniających intonację, akcent oraz ekspresję. Natomiast "narrator.device" dokonuje właściwej syntezy dźwięku, bazując na otrzymanych uprzednio fonemach. W porządku, ale po co takie rozdrobnienia? Nie można było umieścić tych procedur w jednym urządzeniu lub jeszcze lepiej w bibliotece? Nie, podział taki jest starannie przemyślany i już go uzasadniam. Wspomniałem, że "translator.library" tłumaczy tekst uwzględniając m.in. akcent. Amiga powstała w USA i tam też została napisana ta biblioteka, dlatego też, mimo tego, że podamy tekst na przykład w języku polskim, "translator.library" przetłumaczy go według angielskich zasad wymowy (głoska "j" zazwyczaj będzie brzmiała jak "dż", "c" jak "k"). Widać więc, że gdyby producent zamknął wszystkie procedury w jedną bibliotekę do końca życia bylibyśmy skazani na język angielski, a tak nic nie stoi na przeszkodzie aby ktoś na przykład z czytelników Kebaba napisał polską wersję "translator.library".

Pobieżna analiza programu pozwala stwierdzić, że składa się on z trzech głównych części:

1. Otwarcie i inicjacja urządzenia oraz biblioteki.
2. Część zasadnicza czyli odwołanie się do systemowych procedur syntezy mowy.
3. Zakończenie pracy i wyjście z programu.

Przyjrzyjmy się bliżej procedurom wspomnianym w punkcie 2, one to bowiem wykonują całą "czarną robotę". Jak już wyżej pisałem najpierw odwołujemy się do "translator.library", a następnie skacząc do podprogramu "SayIt", na podstawie uprzednio zdefiniowanej struktury "IoRequest", pobudzamy komputer do mówienia. Co zawiera struktura "IoRequest"? Oprócz najróżniejszych danych dla samego "narrator.device" znajdują się tam (oznaczone etykietami) wartości, których zmiana daje ogromne pole do eksperymentów. Ich znaczenie, oraz graniczne wartości podaje poniższa tabela:

Etykieta	Zakres dopuszczalnych wartości	Znaczenie
Rate	40 - 400	Prędkość mowy
Pitch	65 - 320	Tonacja głosu
Mode	0 - 1	0 - głos ludzki, 1 - "głos robota"
Sex	0 - 1	0 - głos męski, 1 - głos damski
Volume	0 - 64	Głośność
Frq	5000 - 28000	Częstotliwość głosu

Teraz nie pozostaje już nic innego, jak skopiować z dyskietki "Workbench" na boot'owany dysk do katalogu "DEVS" plik "narrator.device" oraz do katalogu "LIBS" plik "translator.library", przepisać tekst źródłowy programu, wpisać żądany tekst między etykiety "StartString", a "StopString" i w chwilach nudy porozmawiać z Amigą.

Krzysztof Kobus

ciąg dalszy "Copy-Party"

wsze produkty i rzeczywiście było na co popatrzyć. Ogromną sensację wzbudziło również Video-Demo grupy Silence pt. "Global Trash II" z doskonałymi efektami dźwiękowymi Jesper Kyd'a, dowodzące pełnego profesjonalizmu tej grupy. Bardzo odważnym i wzniosłym akcentem było pojawienie się przedstawiciela firmy zajmującej się sprzedażą licencyjnego oprogramowania. Niestety stoisko to nie cieszyło się powodzeniem. A szkoda... Około godziny 17 zaczęły się konkursy. Wszyscy dostaliśmy starannie przygotowane kartki do głosowania. Rozgrywało się ono w następujących kategoriach:

AMIGA
1. Muzyka.
2. Grafika.
3. Demo.

C64
1. Grafika.
2. Demo.



Sztuczki i triki

w okienku CLI

Zapewne niejeden z Was widział kiedyś, jak na ekranie CLI pojawia się kolorowy tekst, często nawet wzbogacony o takie efekty jak pogrubienie czy podkreślenie. Myśleliście wtedy: "no tak... pewnie assembler..., chyba nigdy tego nie zrobię ". O bogowie! Rozwiązanie jest banalniejsze niż sądzilibyście. Jeśli próbowaliście kiedyś obejrzeć jakiś dysk "produkujący" takie specjalne efekty, to chyba domyśliliście się, że cała tajemnica kryje się najczęściej w pliku startup-sequence. Ten bardzo ważny plik znajduje się w katalogu S tej dyskietki. Podczas gdy komputer boot'uje z dyskietki (np. po włączeniu komputera) system operacyjny stara się najpierw odszukać ten plik i wykonać zawarte w nim komendy. Swoją uwagę powinieneś skupić na linii, w której umieszczona jest komenda "type". Pozwala ona wyświetlać na ekranie tekst zapisany wcześniej do pliku (najczęściej przy pomocy edytora tekstu). Teraz całą sztuką jest dowiedzieć się o jaki plik chodzi. Jeśli przykładowo linia ma postać:
Type sys:s/text
to już wiemy, że chodzi o plik umieszczony w katalogu S naszej dyskietki i noszący nazwę "text". Teraz należy jedynie uruchomić odpowiedni edytor tekstu, np. CygnusEd. Wczytujemy tekst do edytora i... wszystko jasne! Tekst przypomina trochę zawartość wyświetlanego ekranu, ale tylko trochę. Wszędzie roi się od dziwnych znaków. Zauważmy jednak pewną regułę: "dziwne" znaki są pogrupowane, a początkiem każdej takiej grupy jest znak "[" wyświetlony w negatywie. Za nim umieszczony jest zawsze znak "[", ale już bez negatywu, potem jakaś liczba, a wreszcie na końcu litera "m". To jest właśnie rozwiązanie naszego problemu! Wystarczy jeszcze zauważyć, że gdybyśmy usunęli wszystkie "dziwne" znaki, to otrzymalibyśmy dokładnie wygląd ekranu ale... bez kolorów. Co więcej- uzyskiwany na ekranie "efekt specjalny" zależy od tego, jaki "magiczny" kod go poprzedza. Jeśli lubisz eksperymentować, to możesz już przerwać czytanie tego artykułu, dla mniej pracowitych gotowe rozwiązanie:
Aby stworzyć własny ekran z "efektami specjalnymi" zbuduj go przy pomocy edytora tekstu (CED). Następnie w miejscu, w którym chcesz wywołać "efekt specjalny" wprowadź odpowiednią sekwencję kodową. W miejscu zakończenia "efektu specjalnego" wpisz specjalną sekwencję ustawiającą standardowy wygląd tekstu.
Wskazówka: znak "[" w negatywie jest określany jako "control [", "^ [" lub "ESC". Możesz go uzyskać na ekranie edytora CED wciskając klawisz ESC.(Dalej w opisie zamiast ESC piszemy ^ [)



A oto zestaw niezbędnych sekwencji kodowych:
^[[0m - normal (ustawia "zwykły" wygląd tekstu
^[[1m - bold (wytłuszczenie)
^[[2m - text color 2 (ustawia kolor tekstu na 2)
^[[3m - italic (pochyłe pismo)
^[[4m - underline (podkreślenie)
^[[7m - inverse (negatyw)
^[[30m - text color 0 (ustawia kolor tekstu na 0)
^[[31m - text color 1 (ustawia kolor tekstu na 1)
^[[32m - text color 2 (ustawia kolor tekstu na 2)
^[[33m - text color 3 (ustawia kolor tekstu na 3)
^[[40m - back color 0 (ustawia kolor tła na 0)
^[[41m - back color 1 (ustawia kolor tła na 1)
^[[42m - back color 2 (ustawia kolor tła na 2)
^[[43m - back color 3 (ustawia kolor tła na 3)

Teraz pozostaje mi jedynie życzyć Wam udanej zabawy!

Krzysztof Moroń

Tutaj znów małe kłopoty organizacyjne. Do konkursu na najlepszą muzykę zgłosiło się 36 kandydatów. Przyjmując przeciętnie za czas przesłuchania jednej muzyki 4 minuty otrzymujemy około 2.5 godziny ciągłej emisji. Dlatego chyba od 20 pozycji przed głośnikami siedziało dosłownie kilka osób - najprawdopodobniej tylko autorzy, reszta towarzystwa rozpierzchła się po szkole. A oto lista laureatów:
Amiga-muzyka: Raf.
Amiga-grafika: Rys.
Amiga-demo:
1 - Deform
2 - Alchemy
3 - WFMH.

Konkursy na C-64 nie zostały rozegrane z powodu małej liczby ewentualnych uczestników.

Z nieoficjalnych źródeł wiadomo tylko, że demem zasługującym na wyróżnienie było "More than NOPs"

Krzysztof Kobus
Marcin Orłowski

64

Digi-Drummer 64

czyli duża perkusja z małego komputera

Wiadomo nie od dziś, że komputery są bardzo dokładne jeżeli chodzi o odliczanie czasu. Znajduje to wykorzystanie w wielu dziedzinach, jak np. w muzyce, pomiarach itp. My dziś zainteresujemy się wykorzystaniem komputera jako prostego automatu perkusyjnego. Prezentowany tutaj program odtwarza 3 podstawowe dźwięki perkusyjne o brzmieniu w stylu „techno", które odtwarzane są na zasadzie samplingu (czyli tzw. digitalizacji dźwięku). Taki system odtwarzania zapewnia bardzo realistyczny odsłuch naszej perkusji.

Program przeznaczony do „wklepania" zawiera edytor, sample (zapisane cyfrowo dźwięki) i przykładowy rytm. Całość należy wpisać używając dowolnego debuggera lub Korektora. Po poprawnym wpisaniu należy nagrać gotowy Kebab-Drummer na taśmę lub dysk w/g adresów podanych przy listingu jako lokacja programu. Obsługa programu jest dość prosta. Po uruchomieniu ukazuje się naszym oczom ekran roboczy, na którym można edytować nasze własne sekwencje perkusji. A teraz trochę o klawiszach: F1- uruchomienie sekwencji od początku; F3- zatrzymanie odtwarzania; F4- kontynuacja odtwarzania po jego zatrzymaniu; S- zmiana prędkości odtwarzania sekwencji; RETURN- przejście z edycji track'u do edycji pattern'ów i z powrotem. Podczas edycji pattern'u można użyć klawiszy „+" i „-" do zmiany aktualnie definiowanej sekwencji. Zapis track'u zawiera numery poszczególnych pattern'ów w kolejności ich odtwarzania. Zapis zakończony powinien być wartością „FF" w przypadku, gdy życzymy sobie powtarzania całego zapisu od początku lub wartością „FE", gdy chcemy zakończyć odtwarzanie po osiągnięciu przez sekwencer ostatniej pozycji. Mam nadzieję, że przedstawiony program będzię źródłem miłej zabawy dla tych, którzy lubią się trochę pobawić ze swoim komputerem w „elektroniczną perkusję".

Paweł „Polonus" Sołtysiński

Uzupełnienie do Superkopiera z nru 3

Dla wszystkich zasmuconych lub zezłoszczonych posiadaczy drive'u 1541-II, którzy wykonali sobie (POPRAWNIE!) "parallel cable" opisany w poprzednim numerze KEBAB'a i mimo to nie uzyskali pożądanych efektów, małe uzupełnienie. Wkrótce po zamknięciu numeru trafiła do nas stacja 1541-II, która mimo prawidłowego wykonania wszystkich połączeń, nie chciała współpracować ani z naszym programem kopiującym, ani z żadnym innym programem pracującym w oparciu o "parallel cable". Dokładniejsza analiza konstrukcji płyty kontrolera wyjaśniła sprawę. Okazało się, że jest to trochę inna wersja od tych, które spotykaliśmy dotychczas. Co wobec tego mają zrobić wszyscy nasi Czytelnicy, którym trafiła się właśnie ta wersja? Sprawa jest stosunkowo prosta, wymaga jednak ponownego demontażu całego napędu. Zaznaczam po raz wtóry, że dotyczy to tylko tych spośród Czytelników, którzy posiadają stację 1541-II (nowa obudowa, oddzielny zasilacz) i są PEWNI, że wykonali wszystkie połączenia zgodnie z podanym schematem a mimo to nie udało im się uruchomić "SuperKopiera". Zatem bierzemy jeszcze raz do ręki poprzedniego KEBAB'a i demontujemy wszystko ponownie według podanego tam wzoru (chyba, że jeszcze nie zmontowaliśmy naszego drive'u po ostatnich próbach). Po odłączeniu wtyczek od płyty kontrolera i wyjęciu właściwego napędu, musimy wykręcić jeszcze trzy śrubki mocujące płytę do dolnej części obudowy. Wyjmujemy delikatnie płytę uważając jednocześnie na niezbyt mądrze (bo na stałe) przymocowane przewody biegnące do znajdujących się na przedniej części obudowy drive'u diód luminescencyjnych i ew. w ten sam sposób mocowane nasze przewody. Następnie odwracamy płytę kontrolera drugą stroną do góry i odnajdujemy punkty lutowania "naszego" układu 6522 od spodu. Teraz zainteresują nas jego nóżki nr 1 i nr 2. Nożka nr 1 jest zawsze podłączona do masy. Jeżeli mieliśmy pecha to mamy akurat taką wersję, gdzie nóżka nr 2, czyli normalnie bit zerowy nie wykorzystanego portu w układzie 6522, też jest zwarty do masy. W praktyce oznacza to, że pomiędzy punktami lutowniczymi (od spodu płyty) tych nóżek, biegnie cieniutka ścieżka drukowana. Jeżeli udało nam się odszukać w/w ścieżkę, to po upewnieniu się, że jest to na pewno ścieżka drukowana biegnąca pomiędzy dwoma sąsiednimi punktami lutowniczymi znajdującymi się od spodu płyty i należącymi do nóżek numer 1 i 2 układu scalonego 6522 (U6), bierzemy do ręki skalpel (nóż do tapet lub coś podobnego) i delikatnie (ale skutecznie) przecinamy ją. Po wykonaniu tego drobnego zabiegu chirurgicznego, możemy już ponownie zmontować naszą stację uważając oczywiście aby nie uszkodzić niczego co już wykonaliśmy wcześniej (np. nie odłamać przylutowanych przewodów).

SD!

Listy do redakcji

W odpowiedzi na artykuł pióra pana Krzysztofa Kobusa "Dlaczego wolę Amigę" (KEBAB 1/92) chciałbym najpierw podziękować mu za głos w dyskusji. Uważam jednak, że autor się nieco zagalopował w wyciąganiu wniosków i stąd moja obawa czy tym samym nie wprowadził w błąd Czytelników. Wnioski co do których mam pewne zastrzeżenia przedstawiam poniżej.

Po pierwsze:
Zgodnie z twierdzeniem zawartym w Pana artykule, Amiga miała z założenia wyprzeć IBM; ponieważ nic z tego nie wyszło w zamian przekwalifikowano ją na sprzęt domowy. Skoro wypieranie IBM się nie udało, to może warto byłoby porozważać głębiej DLACZEGO tak się stało; nie od rzeczy będzie również refleksja czy ktokolwiek przy zdrowych zmysłach atakowałby światowy standard komputera OSOBISTEGO za pomocą nowego, lecz nieznanego komputera DOMOWEGO.

Po drugie:
Twierdzenie iż "bardzo trudno porównywać PC i Amigę, głównie ze względu na zupełnie inny krąg użytkowników, których wymagania i oczekiwania stawiane tym maszynom są zupełnie odmienne" uważam za mocno naciągane. Wiadomo ogólnie, że tylko PC jest w stanie konkurować z Amigą na polskim rynku, tak pod względem możliwości jak i ceny. W tej sytuacji nie bardzo rozumiem Pana stwierdzenie "porównywanie PC z Amigą nie ma sensu". Jak to? Przecież Amiga była wedle pańskich słów przygotowane z myślą o wyparciu i zastąpieniu IBM, co sam pan napisał w swoim artykule! Jeśli Amiga w żadnym wypadku nie powinna być porównywana z pecetem, to proszę mi powiedzieć z jakim komputerem należy ją porównywać; czekam na Pana sugestie w tej sprawie.

Po trzecie:
Proszę o wyjaśnienie czym właściwie różnią się wymagania użytkownika Amigi od wymagań użytkownika PC. Sądząc po liście oprogramowania jaką pan przytoczył, mam wrażenie, że tak cele jak i wymagania użytkowników obu tych komputerów są zbieżne. W przeciwnym wypadku nikt nie zadawałby sobie trudu, aby robić "wierny amigowski odpowiednik Lotusa 1-2-3" bądź programy typu Dos2Dos. Niech Pan odpowie z ręką na sercu: czy możliwość wymiany plików z pecetami nobilituje Amigę czy też PC?

Po czwarte:
Uważam, że wszelkie sugestie związane z doradzaniem komukolwiek jaki komputer kupić powinny być pozbawione emocji, od których niestety nie był wolny Pana artykuł. Wydaje mi się, że naszym zadaniem jest OBIEKTYWNE informowanie Czytelników o dostrzeżonych tak zaletach jak i WADACH sprzętu - w dzisiejszych czasach zachwyt powinien być również poparty szeregiem faktów.

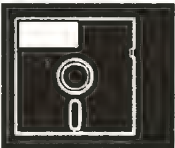
Po piąte:
Deklaruję niniejszym chęć napisania wraz z panem artykułu (lub serii artykułów) poświęconego oprogramowaniu dla PC i Amigi z uwzględnieniem niezbędnej konfiguracji komputera. Myślę, że taki materał będzie przydatny dla Czytelników mających dokonać niebawem wyboru sprzętu.

Pozostaję z poważaniem - Klaudiusz Dybowski

Korzystając z okazji chciałbym bliżej sprecyzować swoje zdanie, dotyczące poruszanego problemu, według zaproponowanej przez Pana Dybowskiego konwencji.

Po pierwsze, drugie i trzecie:
Twierdzenie iż "Wiadomo ogólnie, że tylko PC jest w stanie konkurować z Amigą na polskim rynku, tak pod względem możliwości jak i ceny." nie jest wcale dla mnie tak oczywiste jak dla jego autora. Przede wszystkim nie rozumiem czym różni się nasz rodzimy rynek rozpatrywany na tej płaszczyźnie, od rynków innych państw. Zostawmy może jednak na boku te swobodne dywagacje i wróćmy do sedna sprawy. Jak już wspomniałem w swoim artykule uważam, że Amigę 500 należy zaliczyć do klasy komputerów domowych. Osobom nie będącym przekonanym o słuszności takiej klasyfikacji, pragnę przytoczyć wypowiedź szefa firmy "Electronic Arts" pana Trip Hawkinsa opublikowaną w listopadowym numerze amerykańskiego magazynu "Compute!" z roku 1985, czyli niedługo po premierowej prezentacji Amigi. Aby nie zostać posądzonym o nieprawidłowe tłumaczenie pozwolę sobie wspomniany tekst przytoczyć również w wersji oryginalnej. "The Amiga will revolutionize the home computer industry. It's the first home machine that has everything you want and need for all the major uses of a home computer, including entertainment, education and productivity" Czyli: "Amiga zrewolucjonizuje środowisko komputerów domowych. Jest pierwszym komputerem **domowym**, posiadającym wszystko to co chcesz i potrzebujesz w większości zastosowań domowych, włączając rozrywkę, edukację oraz produktywność.". Zatrzymajmy się przez chwilę i zastanówmy kim będą użytkownicy komputerów domowych... Będą to przede wszystkim pociechy swoich rodziców traktujący Amigę jako zabawkę, która ze względu na jej niecodzienność będzie cieszyć dłużej niż klocki lub elektryczna kolejka, aby być może za pewien czas wyzwolić w dziecku większe ambicje, niż łamanie kolejnych joysticków. Będą to zapaleni koderzy realizujący się w tworzeniu programów demonstracyjnych i prostych programów użytkowych, by kiedyś, w przyszłości stać się "prawdziwymi" programistami już prawdopodobnie na maszynach o większych jeszcze możliwościach. Będą to artyści dający upust swym twórczym żądzom używając Amigi jako pędzla i farby, czy też najbardziej uniwersalnego instrumentu muzycznego jaki kiedykolwiek wymyślono. Będą to wreszcie "dojrzali" użytkownicy, którzy dokończą swoją zawodową pracę wieczorem w domu, aby jej wyniki zanieść zapisane na dyskietce do biura, gdzie znajduje się ich codzienne narzędzie pracy: pecet, korzystając z tego, że ktoś zadał sobie jednak trochę trudu, aby zrobić wierny amigowski odpowiednik Lotusa 1-2-3 i programy typu Dos-2-Dos. Wiadomo również, że nikt obecnie nie kupi Amigi do biura, choćby ze względu na królujący tam standard IBM, notabene którego sukces wywodzi się głównie dzięki bardzo sprytnej i przemyślanej kampanii promocyjnej (osoby zorientowane doskonale powinny wiedzieć o całkowitym udostępnieniu dokumentacji technicznej przez firmę IBM komputera XT, co w ówczesnych czasach mogło mieć i rzeczywiście miało łatwe do przewidzenia skutki), chociaż nie przeczę, że komputery PC w pełni wywiązują się ze stawianych im zadań **biurowych**. Mimo wszystko nie polecał bym jednak zakupu takiego komputera z myślą o zastosowaniach domowych. Dlaczego? Myślę, że na to pytanie najlepiej odpowie cytat z artykułu Pana Marka Sawickiego umieszczonego w tym samym Bajtku, co artykuł Pana Dybowskiego (7/1991). "Każdy gracz wie, co to jest joystick. W przypadku PC-eta korzystanie z tego urządzenia nie jest takie proste, jak mogłoby się zdawać... Wśród programów na IBM PC wcale niemało jest gier. Czasem zdarza się, że gra pozwala na używanie joysticka. W takim momencie zadowolony gracz ściąga z półki pierwszy lepszy joystick, chwyta za wtyczkę i... z reguły żadne gniazdko w IBM-ie do niej nie pasuje. Nie dość tego - gra wspomina coś o joysticku analogowym i o jego kalibracji. Jedyne co w takiej sytuacji pozostaje, to położyć uszy po sobie i grać przy użyciu klawiatury". Wydawać by się mogło, że taka banalna rzecz - podłączenie joysticka, a jednak może sprawić tyle kłopotów. Czy oznacza to, że PC jest gorszy? Oczywiście że nie. Po prostu z założenia miał służyć do czegoś innego - nie do gier. Czytelnikom pozostawiam wnioski do wyciągnięcia, a Panu Dybowskiemu proponuję porównać Amigę do ATARI ST. Uważam, że takie zestawienie jest bardziej miarodajne.





Po czwarte:
Celem mojego artykułu w żadnym wypadku nie było dawanie komukolwiek rad dotyczących kupna komputera. Po prostu był on odzewem na Pański artykuł zamieszczony we wspomnianym już numerze "Bajtka", który też podejrzewam miał inny charakter (osobiście zupełnie inaczej wyobrażam sobie artykuł doradzający kupno sprzętu). Ponadto uważam, iż zarzut braku obiektywności skierowany powinien być przede wszystkim do jego autora. Po ponownym przeczytaniu bajtkowego artykułu stwierdziłem, że poza wzmianką "(...) Amiga daje użytkownikowi znacznie większe możliwości graficzne i dźwiękowe aniżeli standardowy PC" nie znalazłem wśród ogromu opisanych tam wad, żadnej jej cechy pozytywnej. Czyżby Amiga była komputerem bez zalet? Jak zatem wytłumaczyć miliony już sprzedanych jej egzemplarzy?



Po piąte:
Jako redaktor Kebaba odpowiedzialny jestem za zawartość merytoryczną publikowanych w nim artykułów. Z osobiście przeprowadzonych ankiet wiem, że Czytelników znacznie bardziej interesują materiały innego typu, na przykład dotyczące programowania w assemblerze - stąd cykl "Mapa pamięci Amigi". Biorąc pod uwagę bardzo małą ilość literatury poruszającej tego typu sprawy dotyczące komputerów PC, proponuję Panu stworzyć na łamach Bajtka analogiczny cykl artykułów dotyczących programowania kart graficznych, procedur systemu operacyjnego, lub sprzętowych rejestrów specjalizowanych układów komputera.

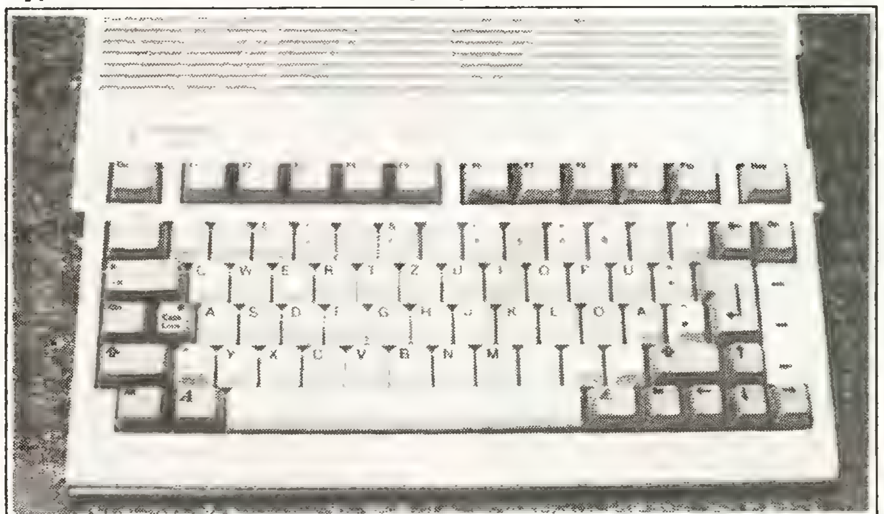
Krzysztof Kobus



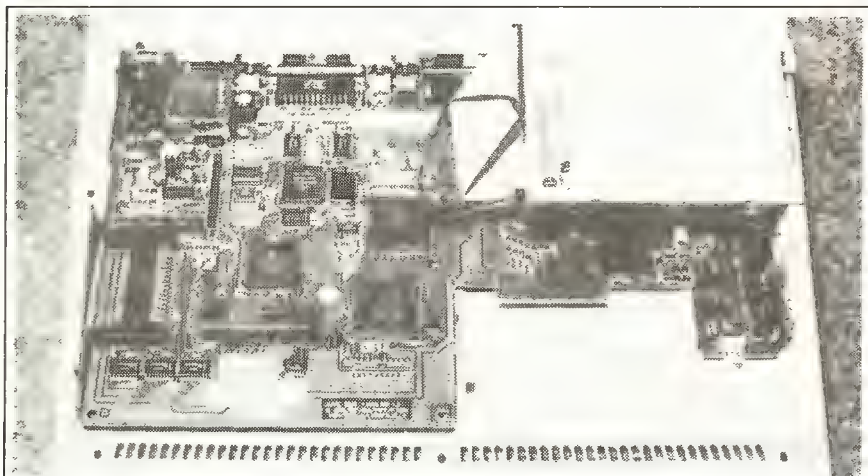
Amiga 600

i co się za tym kryje

Kolejną niespodzianką, jaką przygotowali nam specjaliści - konstruktorzy firmy Commodore jest zaprezentowana po raz pierwszy na targach CEBIT '92 Amiga A-600 i A-600HD. HD to po prostu skrót od Hard Drive czyli twardy dysk. Co to jest właściwie takiego ta Amiga 600? Nowy komputer? Nowa maszynka do gier? Czy może jeszcze coś innego? Muszę stwierdzić, że urządzenie to budzi we mnie bardzo mieszane uczucia. Z jednej strony jest to pierwsza poważniejsza zmiana w konstrukcji Amigi począwszy od 1986 roku a z drugiej nie bardzo mogę jednoznacznie stwierdzić czy jest to krok naprzód, czy do tyłu. Pozostaje mieć nadzieję, że nie jest to ostatni etap w rozwoju konstrukcyjnym Amigi (zwłaszcza jej mniejszych modeli). Pierwsza sprawa widoczna po rozpakowaniu komputera to zupełnie inny kształt obudowy. Czegoś nam przy okazji jakby brakowało... AHA! Brakuje klawiatury numerycznej. Co prawda do „giercowania" była ona mało przydatna, ale niektóre gry również z niej korzystały. Na pewno natomiast była ona przydatna do innych zastosowań. Dobrze, było i minęło. Nowa Amiga jest m. in. dzięki temu znacznie mniejsza. Z prawej strony pozostała kieszeń stacji dysków z lewej pojawiła się druga, tym razem przeznaczona na nowego typu rozszerzenia pamięci typu „Flash-Memory-Card" mogą one być wyposażone zarówno w pamięć typu RAM jak i ROM. Żeby było ciekawiej dołączono do systemu operacyjnego (37.300) procedury (carddisk.device i card.resource) obsługi tychże kart na podobnych zasadach co normalnych dyskietek. Oznacza to m. in. to, że można boot'ować nową amigę nie korzystając z dyskietki lecz z karty z zawartą na niej np. grą. Na pytanie: Czy moje wszystkie drogie urządzenia dodatkowe, które dokupiłem sobie do A-500 będą działały z nową A-600? należy odpowiedzieć: Oczywiście... że nie! Urządzenia wkładane do A-500 od spodu (rozszerzenia pamięci, emulator IBM'a - KCS Power Board itp.) nie będą działały bo... nie ma ich gdzie włożyć. Urządzenia dołączane do A-500 z boku (twardy dysk, pamięć, cartridge, scanner) też nie będą działać z tych samych powodów. Jeżeli chodzi o twardy dysk to z tym jeszcze pół biedy bo wyposażenie A-600 w takowe urządzenie zostało przewidziane przez konstruktorów. W wesji A-600HD firma instaluje od razu 2 i pół calowe napędy z podłączeniem typu IDE-Bus. W „zwykłej" sześćsetce natomiast pozostawiono uchwyty mocujące tak, że ewentualne zainstalowanie takiego napędu we własnym zakresie, pod warunkiem rezygnacji z gwarancji, nie powinno nastręczać wielkich problemów. Po zdemontowaniu tychże uchwytów oraz blachy ekranującej ukazuje nam się wnętrze nowej Amigi w całej okazałości. Tutaj kolejna niespodzianka. Gdyby ktoś chciał zainstalować np. opisywany w tym numerze KEBAB-'a Flicker-Fixer, to może od razu z takich pomysłów zrezygnować. Dlaczego? Dlatego, że wszystkie układy scalone w A-600 za wyjątkiem układu pamięci ROM z systemem operacyjnym są wykonane w tzw. technologii SMT (Surface Mounting Technology = Technologia Montażu Powierzchniowego) i są urządzeniami typu SMD (SM Devices = Urządzenia Montowane Powierzchniowo). W praktyce oznacza to znaczną oszczędność miejsca (SMT pozwala na znacznie gęściejsze upakowanie elementów). Wiadomo już dlaczego A-600 jest prawie o połowę mniejsza od „pięćsetki". SMT pozwala również na znaczne obniżenie kosztów produkcji. Ma natomiast jedną zasadniczą wadę. Wymiana jakiegokolwiek elementu (nie mó-



Amiga 600 z zewnątrz...



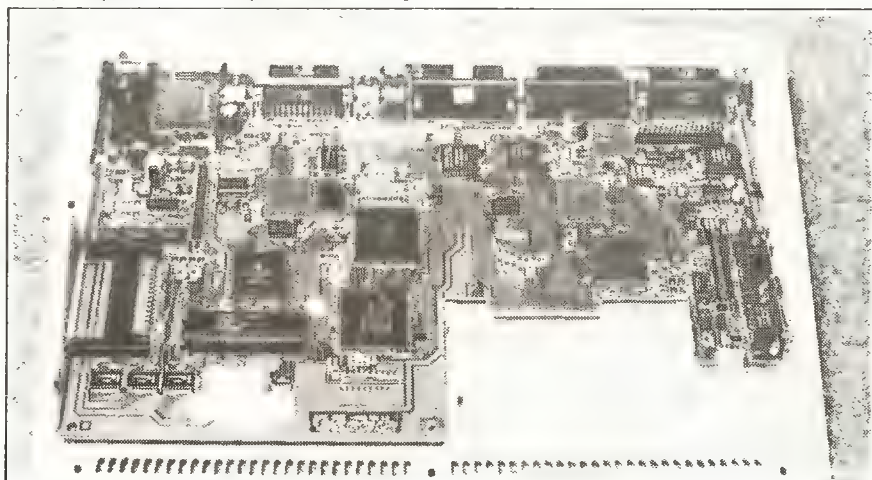
... i od środka

wię tu już o instalowaniu różnych dodatkowych urządzeń) jest praktycznie rzecz biorąc nie do zrealizowania w warunkach domowych. Co zatem z ciągle psującymi się układami FAT (BIG, SUPER BIG) AGNUS czy portami wejścia/wyjścia (I/O) typu 8520? Nic! Zostały one również wykonane jako SMD co z jednej strony eliminuje możliwość powstawania błędów pracy komputera spowodowanych nieprawidłowo kontaktującymi stykami w dotychczasowych podstawkach, a z drugiej strony każdą ewentualną naprawę sprowadza (oprócz ceny samej części) do bardzo kosztownego wylutowywania i ponownego lutowania układów w odpowiednim warsztacie. Ewentualni nabywcy powinni mieć to na uwadze. Co poza tym ciekawego w tym (?) komputerze? Gniazda joystick'ów (ew. myszki) zostały według starych, dobrych wzorców (C-64) przesunięte na prawą stronę obudowy. Z tyłu przybyło nowe gniazdo do podłączenia... telewizora! Tak! Modulator, dokupowany dotychczas oddzielnie, został zainstalowany na płycie głównej. Oprócz tego z tyłu obudowy znajdują się wszystkie znane z A-500(plus) gniazda: parallel, serial, disk drive, audio itd. Zgodnie z zapewnieniami firmy Commodore, A-600 ma być w pełni kompatybilna z A-500plus. Posiada więc również 1MB CHIP-Ram, cały ECS czyli nowa (Hires) Denise oraz (SUPER BIG) Agnus jak i Kickstart 2.0 (dokładnie: 2.05), który powinien być bardziej „strawny" dla gier niż dotychczasowy (2.04). Ogólnie trudno jest jednoznacznie ustosunkować się do tego nowego produktu Commodore. O ile swego rodzaju rozterki przeżywaliśmy już podczas testu A-500plus (KEBAB 1'92) o tyle były to głównie rozterki typu: Dlaczego tak mało ulepszono? W przypadku A-600 natomiast pytanie które sobie ciągle zadaję spoglądając na nią brzmi: Co to właściwie jest? Modulator? No tak, maszynka do gierek... Ale! Twardy dysk od razu w środku? To sugeruje poważniejsze zastosowania. Znowu możliwości rozbudowy tego systemu ograniczone są praktycznie do zubożonego (względem A-500) łącza wewnątrz obudowy. SMD w środku? Odpadają problemy z wadliwymi podstawkami, ale i wiele ciekawych możliwości. Problem styków można było przecież rozwiązać poprzez zastosowanie (droższych) podstawek typu „precision", ale kto by chciał ponosić dodatkowe koszty? Ciekawostką jest fakt, który świadczy o tym, że również twórcy A-600 nie bardzo chyba wiedzieli czy wypuszczenie tego modelu na rynek to krok naprzód czy wstecz. Mam tu na myśli niezgodność pomiędzy oznaczeniem handlowym modelu (na obudowie), które brzmi „A-600" a oznaczeniem konstrukcyjnym (na płycie głownej), brzmiącym „June Bug - A300".

+ małe wymiary
+ przewidziana możliwość zainstalowania twardego dysku wewnątrz obudowy
+ gniazdo kart pamięci Flash-Memory-Cards (PCMCIA)
+ modulator

- technologia SMT uniemożliwiająca stosowanie rozmaitych rozszerzeń w postaci „przejściówek" (ATonce, Flicker-Fixer, Fast-RAM itp.)
- brak standardowych dla A-500 złącz
- brak klawiatury numerycznej
- wysoka cena zakupu i ew. napraw

SD!



Pod stacją mamy układy 8520 (SMD)

64

Jak zrobić scrolla

na C-64

Pomimo faktu, iż jest to dopiero trzeci numer KEBAB-'a, możemy się już cieszyć z listów od naszych Czytelników. W kilku z nich prosiliście o wyjaśnienie w sposób przystępny techniki przewijania tekstu na ekranie, czyli tzw. scrollingu. Najczęściej chcielibyście umieścić taką procedurę na ekranie tytułowym, która dodała by mu choćby uroku. Niestety, w języku BASIC uzyskanie pełnej płynności przesuwu jest niemożliwe (posługując się samym tylko BASIC'iem). Pozostaje nam tylko umiejętne napisanie procedury w ten sposób, by pracowała jak najszybciej oraz by była wygodna w użyciu. Kierując się tymi wytycznymi napisałem krótki programik, który realizuje naprawdę szybkiego scroll'a w wyznaczonej linii ekranu. Dzięki zastosowaniu pewnego trick'u jest ona tak szybka, że bez zastosowania pętli opóźniającej bardzo trudno byłoby cokolwiek odczytać. Ale lepiej przystąpmy do szczegółowych wyjaśnień, bo warto to wszystko jeszcze zrozumieć. Przy realizowaniu „płynących literek" w momencie, w którym drukujemy kolejną literę z prawej strony ekranu, należy przedtem przesunąć pozostałe znaki w tej linii o jeden w lewo. W ten sposób pierwszy znak od lewej strony „znika" robiąc miejsce dla nowego, który, jak już wspomnieliśmy, wstawić należy po prawej stronie. Głównym problemem, który trudno jest zrealizować w BASIC-'u, jest właśnie owo przesunięcie znaków o jeden w lewo. Napisanie przez nas np. procedury, która znaki te przesunie po kolei o jedną pozycję w lewo mija się z celem, gdyż BASIC jest do tych celów za wolny, a poza tym, tak napisany podprogram trzymał by się sztywno konkretnej linii, dla której został napisany. Wyjściem okazało się skorzystanie z właściwości edytora ekranowego, którym posługujemy się przy pisaniu programów w BASIC. Łatwo zauważyć, że podczas użycia klawisza DEL taka właśnie interesująca nas forma przesuwania liter o jeden znak w lewo występuje (zwróćcie uwagę na to, co się dzieje ze znakami położonymi na prawo od kursora). Procedury umieszczone w ROM komputera przesuwają nie tylko znaki ale także odpowiadające im atrybuty kolorów, dzięki czemu nasz scroll może być napisany z użyciem kolorów: wystarczy użyć tylko odpowiednich kodów sterujących w tekście. Przystąpmy teraz do opisu samego programu: w linii o numerze 15 znajduje się definicja zmiennej W, która wskazuje procedurze, w której linii ma być realizowane przewijanie tekstu (licząc od góry ekranu). W linii 20-tej znajduje sie tekst, który będzie przesuwany. Został on przyporządkowany zmiennej TEKST$. Dla przypomnienia - maksymalna długość tekstu wynosi 256 znaków, gdyż tyle maksymalnie może wynosić długość łańcucha znaków w Commodore Basic. W linii 30 następuje czyszczenie ekranu i ustalenie wiersza roboczego na podstawie zmiennej W. Jeżeli z jakichś względów nie chcemy kasować zawartości ekranu, wystarczy zmienić wartość CHR$(147) na CHR$(19). Począwszy od linii 40 następuje właściwa procedura przesuwu liter: linia 40 to właśnie przesunięcie znaków o jeden w lewo z wykorzystaniem DEL (PRINT CHR$(20); linia 50 to pobranie kolejnego znaku ze zmiennej TEKST$; linia 60 to wydrukowanie tego znaku; linia 70 to pętla opóźniająca (proszę trochę poeksperymentować z parametrem „60") i wreszcie w linii 80 następuje testowanie klawiatury - jeżeli wciśnięta została spacja to program kończy swoją pracę i powraca do trybu edycji.

Paweł „Polonus" Sołtysiński

llisting nr 3 na stronie 31



Guru-meditation

Zapewne niejednokrotnie pracując na swoich Amigach, szczególnie w czasie tworzenia własnych programów, w górze ekranu ujrzeliście czerwoną, migającą ramkę z napisem „Guru-Meditation". Jest to niewątpliwie najstraszniejszy widok jaki może ujrzeć użytkownik tego komputera. Nie wszyscy jednak wiedzą, iż ramka ta, a właściwie cyfry w niej zawarte, niosą użytkownikowi bardzo ważne

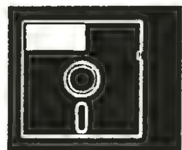
informacje, które pozwalają dociec jaki błąd został popełniony. Znając ich kody możemy łatwo rozszyfrować informacje jakie niesie za sobą Guru i w stosunkowo prosty sposób poprawić nasz program.
Guru-Meditations dzielą się na dwa rodzaje:

I. Błędy procesora.
Jeżeli zaistnieje tego rodzaju błąd, system zgłosi informację zapisaną w formacie:
Guru Meditation
#0000000a.bbbbbbbb
gdzie:
bbbbbbbb Jest adresem task'u którego dotyczy błąd,
a Jest jedną z następujących wartości:
2 - błąd magistrali
3 - błąd adresowania (np. gdy przesyłamy „word" od nieparzystego adresu)
4 - nielegalna instrukcja (występuje po napotkaniu instrukcji „ILLEGAL")
5 - dzielenie przez zero
6 - instrukcja CHK
7 - instrukcja TRAPV
8 - narzucenie uprzywilejowania
9 - śledzenie
A - emulator linii A (instrukcje o kodach $A000-$AFFF)
B - emulator linii F (instrukcje o kodach $F000-$FFFF)

II. Błędy programowe.
Błędy tego rodzju posiadają następujący format:
Guru Meditation
#aabbcccc,dddddddd
gdzie:
aa Wskazuje bibliotekę (ang: library), urządzenie (ang: device), itp., którego błąd dotyczy. Poza tym przekazuje informację czy błąd był na tyle poważny, aby spowodować całkowite załamanie systemu, czy też nie. Możliwe są następujące przypadki:
Biblioteki:
01 - exec
02 - graphics
03 - layers
04 - intuition
05 - math
06 - clist
07 - dos
08 - ram
09 - icon
0A - expansion
Urządzenia:
10 - audio
11 - console
12 - gameport
13 - keyboard
14 - trackdisk
15 - timer
Resources:
20 - cia
21 - disk
22 - misc
Inne:
30 - bootstrap
31 - workbench
Jeżeli wartości wyżej podane będą zwiększone o 80, oznacza to, że błąd spowodował całkowite załamanie systemu.
bb Podaje rodzaj błędu.
01 - Brak pamięci
02 - Błąd podczas tworzenia biblioteki
03 - Błąd podczas otwierania biblioteki
04 - Błąd podczas otwierania urządzenia
05 - Błąd podczas otwierania resource'a
06 - Błąd wejścia/wyjścia
07 - Brak sygnału
cccc Podaje szczegółowe informacje o błędzie. Pole to ma różne znaczenie dla różnych podsystemów, dlatego przedstawię teraz przykładowe, najczęściej spotykane wartości wraz z ich objaśnieniami.
exec:
0001 - zła suma kontrolna wektora stanu wyjątkowego 68000
0002 - zła suma kontrolna ExecBase
0003 - błąd sumy kontrolnej biblioteki
0004 - brak pamięci na wykreowanie biblioteki
0005 - zniszczona lista pamięci
0006 - brak pamięci dla obsługi przerwania
0009 - próba dealokacji pamięci nie zaalokowanej
000A - niezaimplementowany stan wyjątkowy
graphics:
0000 - brak pamięci dla graphics
000A - brak pamięci dla BltBitMap
0030 - brak pamięci na utworzenie ViewPort'u
1234 - pamięć nie dostępna w sytuacji krytycznej
intuition:
0001 - gadget nie znanego typu
0002 - brak pamięci na wykreowanie portu
0007 - brak pamięci na otworzenie screen'a
0008 - brak pamięci na alokację pamięci obrazu dla screen'a
0009 - próba otwarcia screen'a nie znanego typu
000A - brak pamięci na dodanie gadget'ów
000B - brak pamięci na otwarcie okna
000D - zła wiadomość (ang: message) przybyła przez kanały IDCMP
000F - błąd przy otwieraniu console.device
layers:
0000 - brak pamięci dla layers
dos:
0002 - brak możliwości wykonania EndTask
0004 - przybycie nieoczekiwanego pliku danych
0007 - zniszczona mapa sektorów zaalokowanych
0009 - zła suma kontrolna
000A - błąd przy próbie wykonania operacji dyskowych
000C - zła nakładka
dddddddd Może mieć trojakie znaczenie:
* zawiera adres task'u zgłaszającego błąd,
* jeżeli błąd nastąpił podczas operacji alokacji/dealokacji pamięci zawiera adres tego bloku pamięci,
* w przypadku, gdy nastąpił bardzo poważny błąd zawiera 48454C50, co odpowiada kodom ASCII słowa HELP. W ten właśnie sposób przedstawiają się informacje zawarte w Guru-Meditations. Mam nadzieję, że od dnia dzisiejszego nie będą one tylko tajemniczymi liczbami, lecz pomogą Wam w poprawianiu nieuniknionych (zgodnie z prawami Murphi'ego) błędów, których życzę jak najmniej.

Krzysztof Kobus





Hudson Hawk

zamiast opisu...

W wielu magazynach poświęconych tematyce komputerowej, w tym także w Kebabie, można znaleźć działy przeznaczone dla miłośników wszelkiego rodzaju gier komputerowych. Zamieszczane są tam zwykle opisy oraz podpowiedzi do gier a także różnego rodzaju Hints'n'-Tricks czyli rozmaite sztuczki pozwalające graczowi na zwiększenie szans na ukończenie danej gry. Są to jednak zawsze tricki pozostawione celowo przez autorów danego programu (np. w celach testowych). Co jednak robić, gdy mimo usilnych starań, w żadnym piśmie nie możesz znaleźć owych tricków dla swojej ulubinej gry, a na widok napisu GAME OVER dostajesz już drgawek? W takim przypadku proponuję przejęcie inicjatywy w swoje ręce oraz naprawienie „błędów" i „niedopatrzeń" pozostawionych w grze przez autorów, które skutecznie uniemożliwiają ukończenie jej w rozsądnym czasie. W tym celu należy uzbroić się w trochę wiedzy na temat programowania w assemblerze, w dobry monitor (najlepiej z opcją disassemblacji w obie strony i możliwością poszukiwania relokowalnych adresów), oraz w dużo czasu i cierpliwości. Po wysłaniu rodziny na długi spacer lub do kina (najlepiej na wszystkie możliwe seanse) możemy się zabrać do dzieła. Na początku jednak muszę rozczarować niektórych Czytelników pewną uwagą: nie ma żadnej uniwersalnej metody, którą można by stosować przy modyfikacjach w odniesieniu do wszystkich gier. Czasami wystarczy godzina i mamy program „poprawiony", czasami zaś nawet po tygodniu można nie posunąć się nawet na krok. Istnieją jednak dwie drogi w „rozgryzaniu" programów, które mogą być najbardziej przydatne tzn: analiza programu krok po kroku - metoda dość żmudna i czasochłonna, lecz często prowadzi do różnych ciekawych odkryć oraz metoda tzw. „tylnego wyjścia" której użycie mam zamiar szerzej tu opisać.

Przed przystąpieniem do przerabiania programu należy się mu dokładnie przyjrzeć, zobaczyć jaki mamy limit „żyć" w grze, jakie napisy pojawiają się w przypadku porażki, czy są używane dodatkowe klawisze, czy mamy limit czasu itd. Wszystko to może być bardzo pomocne w późniejszej pracy. Dla przykładu weźmy grę Hudson Hawk firmy Ocean opartą na filmie o tym samym tytule. Jest to dość przyjemna gra zręcznościowa, niestety w jej ukończeniu przeszkadzają nam różni strażnicy, psy, odbierający cenną energię, zaś początkowe 4 „życia" to raczej za mało by dojść do ostatniego, piętnastego poziomu. Do tego wszystkiego na przejście każdej strefy wyznaczony jest czas, którego jak zwykle jest zbyt mało. Wczytujemy monitor, a z jego pomocą główny blok programu. Jak łatwo zauważyć jest on kompresowany za pomocą DefJam Packer'a i najpierw musimy go rozpakować. W tym celu należy znaleźć pierwszy rozkaz JMP, zamienić go na rozkaz RTS (lub odpowiednią komendą monitora założyć w tym miejscu tzw. pułapkę (breakpoint), i uruchomić go pominąwszy instrukcje wyłączające przerwania. Jeżeli wszystko przeprowadziliśmy prawidłowo to na ekranie powinny pokazać się kolorowe pasy świadczące o rozpakowywaniu gry, a następnie, po napotkaniu wpisanego przez nas uprzednio rozkazu RTS (lub breakpoint'a) sterowanie powinno zostać przekazane do monitora. Teraz należy odczytać adres pod jaki skakał wspomniany już rozkaz JMP. Będzie on początkiem naszej gry i należy go zapamiętać. Analizując program od miejsca uruchomienia znajdujemy (UWAGA: Intro pomijamy!!!) procedurę relokującą główną część programu. Wygląda ona jak w poniższej tabeli. Uwaga!!! Ze względu na możliwość istnienia wersji gry łamanych przez różnych hacker'ów adresy tu wymienione mogą się różnić.

```
.042326  41FA  020A      lea      $00042532(pc),A0
.04232A  2B48  00E0      move.l   A0,$00E0(A5)
.04232E  2B48  00E4      move.l   A0,$00E4(A5)
.042332  2B48  00E8      move.l   A0,$00E8(A5)
.042336  2B48  00EC      move.l   A0,$00EC(A5)
.04233A  2B48  00F0      move.l   A0,$00F0(A5)
.04233E  2448            movea.l  A0,A2
.042340  93C9            suba.l   A1,A1
.042342  D5FC  00014000  adda.l   #$00014000,A2
.042348  22D0            move.l   (A0),(A1)+
.04234A  4298            clr.l    (A0)+
.04234C  B5C8            cmpa.l   A0,A2
.04234E  66F8            bne.s    $00042348
.042350  4EF9  000052E0  jmp      $000052E0
```

tab. 1

Najwygodniej byłoby ją uruchomić i analizować program znajdujący się we właściwym miejscu. Niestety dysponując zwykłym monitorem nie możemy tego zrobić, gdyż program relokuje się po adres $000000 co raczej na pewno spowoduje utratę kontroli nad komputerem. Pozostaje nam więc przeliczanie adresów. Wpierw jednak proponuję nagrać obszar pamięci od adresu znajdującego się po rozkazie LEA (w naszym przypadku: $42532) do adresu powiększonego o $14000 na dyskietkę (abyśmy nie musieli za każdym razem na nowo go decrunch'ować), po czym na nowo załadować do pamięci pod adres, na przykład $40000, aby przeliczanie adresów nie było uciążliwe. Jak widać, rzeczywiste adresy otrzymamy przez dodanie do pokazywanych przez monitor wartości liczby $40000. (Na przykład w rozpatrywanym przykładzie skok pod adres $52e0 będzie oznaczał dla nas skok pod $452e0.) Zacznijmy od nieśmiertelnej „nieśmiertelności". W przypadku straty wszystkich ludzików ukazuje się, zwykle w najmniej oczekiwanym momencie, napis GAME OVER. Musimy odszukać ten napis oraz procedurę która go wyświetla i zobaczyć kiedy jest ona wywoływana. Kody ASCII odpowiadające „gAME oVER" znajdujemy pod adresem $511ac. Teraz szukając odwołania do tego adresu znajdujemy procedurę wyświetlania napisu. Disassemblujemy pamięć „w górę", do czasu napotkania pierwszego rozkazu RTS, JMP, BRA (kończącego zwykle poprzednie procedury), po którym spodziewamy się znaleźć początek procedury GameOver. I rzeczywiście, mamy:

```
.04F996 bsr    $000501F4
.04F99A bra.s  $0004F9B6
```
;pierwszy napotkany BRA
```
.04F99C bsr    $0004979E
```
;początek procedury GameOver
```
.04F9A0 lea    $530C,A4
.04F9A4 lea  $537A,A1
.04F9BC bsr  $000501FC
.04F9C0 bsr  $0004FA82
.04F9C4 lea  $511AC(pc),A5
```
;odwołanie do napisu
```
.04F9C8 moveq #$28,D1
.04F9CA bsr  $00051A28
```

Jak widać początek procedury GameOver znajduje się pod adresem $4f99c (przypominam, że w rzeczywistości jest to adres $f99c!). Poszukiwania skoku pod adres względny $4f99c zawodzą. Musimy więc szukać „prawdziwego" adresu, czyli: $f99C. Pełen sukces:

```
.04DA16 subq.w  #1,$52E8
```
;odejmuje 1 od zawartości $52e8
```
.04DA1A bpl.s $0004DA22
```
;jeśli dodatnia to dalej
```
.04DA1C jmp $0000F99C
```
;jeśli nie to skok do GameOver
```
.04DA22 bra  $0004FC58
```
;kontynuacja

oraz:

```
.045530 tst.w $52E8
.045534 bpl.s $0004553C
.045536 jmp $0000F99C
.04553C bra $000453EA
```

Jak widać program testuje zawartość komórki $52e8, która przechowuje aktualną ilość ludzików, i jeśli jest ona ujemna ($ffff) skacze do procedury GameOver. Jeśli w obu miejscach zmienimy BPL na skok bezwarunkowy BRA to Hudson stanie się nieśmiertelny. Można też inaczej. Poszukajmy odwołań do tej komórki:

```
.04548E subq.w #1,$52E8
```

oraz to samo pod adresem $4da16. Wpisując w te miejsca po dwa rozkazy NOP osiągniemy podobny efekt jak w przypadku zmian rozkazów skoków warunkowych. Szukajmy dalej:

```
.04FA06 addq.w #1,$52E8
```

i wreszcie:

```
.050F7E move.w #$04,$52E8
```

Ostatni z przedstawionych rozkazów wstawia początkową wartość (czyli cztery) do naszej komórki. Nie należy jednak przesadzać ze zmianą tej wartości ponieważ może to doprowadzić do małego bałaganu w górnej części ekranu.

Teraz weźmy się za zegar. Sygnalizuje on koniec czasu niezbyt miłym komunikatem TIMEOUT. Postępujemy analogicznie jak wyżej, czyli znajdujemy pod adresem $51388 tekst 'tIMEOUT', nastepnie pod adresem $4d9f0 znajdujemy odwołanie do niego i parę bajtów wcześniej przed pierwszym napotkanym rozkazem RTS znajduje się początek szukanej procedury:

```
.04D9BE rts
.04D9C0 tst.b $52F4
```
;początek procedury
```
.04D9C4 beq.s $0004DA26
.04D9C6 tst.b $18B4
.04D9CA bne.s $0004DA26
.04D9CC tst.b $14A2
.04D9D0 bne.s $04D9E4
.04D9D2 tst.b $156c
.04D9D6 bne.s $04DA26
.04D9D8 clr.w $53C0
.04D9DC moveq #$0E,D0
.04D9DE jsr $0006F8D2
.04D9E4 jsr $579A
.04D9E8 bsr $0004FAC8
.04D9EC bsr $0004FA82
.04D9F0 lea $51388(pc),A5
```
;odwołanie do napisu
```
.04D9F4 moveq #$28,D1
.04D9F6 bsr $00051A28
```

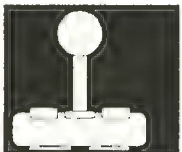


W tym momencie można poeksperymentować z wartościami poszczególnych komórek i obserwować efekty, albo szukać dalej odwołań do tych adresów i sprawdzać ich znaczenie oraz w jakich momentach są zmieniane. Pozostawiam to Czytelnikowi, a teraz podam krótko jedynie do czego sam doszedłem.

Pojawienie się w komórce $52f4 wartości różnej od zera sygnalizuje koniec czasu, zatem prosta zamiana BEQ na BRA rozwiązuje nasze problemy.

$14a2 - wartość różna od zera zatrzymuje zegar (np. w przypadku pauzy). Nie należy jednak tego modyfikować, bowiem przestanie być aktywna opcja „pause" (klawisz P) oraz procedura nabijająca „Time Bonus" zapętli się przy najbliższej okazji.

$156c - jeśli zawiera wartość równą zero to nasz bohater jest widoczny, w przeciwnym wypadku znika z ekranu aż do momentu ponownego wyzerowania tej komórki (jeśli jest niewidzialny nie można nim sterować).



Najciekawsza jest komórka $18b4 i dlatego zostawiłem ją na koniec. Wstawienie do niej w dowolnym momencie podczas gry wartości ujemnej ($ff) powoduje przeniesienie gry do następnego poziomu. Można pokusić sie o wstawienie odpowiedniej procedury reagującej np. na wciśnięcie przycisku myszy i przenoszącej Hudson'a do następnej strefy (problemem może być tylko znalezienie nieużywane-

go obszaru pamięci).
Została nam jeszcze energia (a właściwie jej zbyt mała ilość). Wystarczy zmienić:

```
.04DAD2 subq.b #1,$53A4
.04DAD6 bgt.s $0004DAE4
```

na np:

```
.04DAD2 move.b #$10,$53A4
;wpisywanie początowej
energii .04DAD8 bra.s
$0004DAE4
;skok dalej
.04DADA subq.l #1,-(A4)
;dalsza część procedury
.04DADE st          $53BC
```

Nie należy się przejmować, że nowe rozkazy są dłuższe i zniszczymy fragment kodu. Zmiana BGT na BRA powoduje że i tak nie zostanie on nigdy wykonany. Ta poprawka uodparnia Hudsona tylko na ciosy strażników, psy, wentylatory lub nosorożca w bibliotece (??!), ciągle jednak pozostaje wrażliwy na wysokie napięcie i temperaturę.
Po skończonej pracy, w nagrodę, możemy sobie chwilę pograć. I tu niemiła niespodzianka: zamiast gry ukazuje nam się pusty ekran w czerwonym kolorze. Jak widać, autor gry postanowił zabezpieczyć swoje dzieło przed ingerencją potencjalnych hacker'ów i umieścił w programie odpowiednią procedurkę. Na nasze szczęście nie popisał się w tym przypadku zbyt dużą fantazją bowiem po krótkich poszukiwaniach znajdujemy:

```
.0405DE tst.l D0
.0405E0 beq.s $000405EA
.0405E2 move.w
#$0F00,$15C(A3)
.0405E8 bra.s $000405E2
.0405EA movem.l (A7)+,A0
```

Jak zwykle zamieniamy BEQ na BRA i możemy już bez problemu wprowadzać nasze poprawki. Proste?

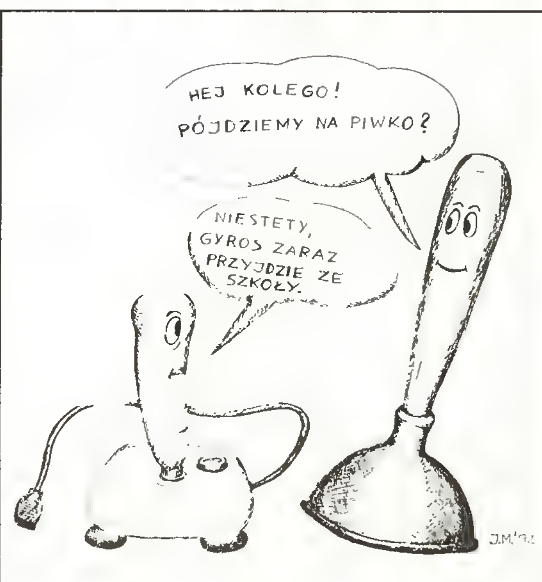
Zamiast zakończenia:
Analiza cudzych programów, pozwala na poznanie różnych chwytów i trick'ow stosowanych w kodowaniu, a przede wszytkim prowadzi do dobrego poznania języka maszynowego, co na pewno zaowocuje w przyszłości przy pisaniu własnych programów, a może nawet gier! I nie zniechęcajcie się jeśli przeróbka nie uda się za pierwszym podejściem. Miłego i owocnego hack'owania życzy:

Marcin „MR SOFT" Orłowski

SIM - CITY

Czy wiecie jak przebiega życie w mieście? Tak, tak, w prawdziwym mieście. Jak powstaje ruch uliczny, dzielnice największego zaludnienia, dzielnice hoteli, banków i ekskluzywnych sklepów. Jest taki program (bo gra to raczej nie jest), gdzie komputer symuluje życie miasta. Program te to właśnie Sim City, czyli Simulated City, co w wolnym przekładzie oznacza Miasto Symulowane. Zabawę z komputerem możemy rozpocząć od zaprojektowania terenu pod nasze przyszłe miasto. Można więc posadzić drzewa, pokryć ląd wstęgami rzek lub nawet zaprojektować brzeg morski. Następnie musimy postawić jakiś obiekt, wokół którego rozpocznie się aktywność miejska - najlepiej sprawdza się w tej roli port wraz z niewielkim zapleczem magazynów i obiektów przemysłowych. Obiekty przemysłowe to miejsca pracy dla ludzi, którzy chętnie by się tu osiedlili, więc na miejscu jest wybudowanie małej grupy domów mieszkalnych. I tak się to powoli zaczyna. Po paru komputerowych latach zaczynają się uwypuklać pierwsze różnice - domy są najchętniej zaludniane (a więc najdroższe) na terenach w dużej odległości od zakładów przemysłowych, najchętniej nad brzegiem morza, blisko parku, z bieżącą wodą itp. Po wybudowaniu lotniska opłaca się tereny obok niego pokryć budownictwem komercyjnym (hotele, banki), co wydajnie zwiększy ilość miejsc pracy oraz np. nasili ruch samochodowy, co z kolei pociąga za sobą konieczność przebudowy dróg i burzenia niektórych budynków. Przyjemnie jest potem obserwować, jak Twoje miasto "żyje". Dodatkowo komputer na nasze życzenie w każdej chwili może nam podać aktualną mapę barwną cen ziemi, nasilenia ruchu na drogach, standardu życia, czy wielkości populacji i stopnia zanieczyszczeń. Kiedy nasze miasto już prosperuje naszym zdaniem aż za dobrze, korzystając z pomocy komputera możemy zesłać na nie parę stosownych wypadków losowych, jak np. trzęsienie ziemi lub fala pożarów, co od razu poprawi nam humor. Zabawa długo się nie nudzi, co jest najlepszą miarą pomysłu. Jedyną wadą jest nie najlepsze wykonanie programu, przy czym mam tu na myśli nie procedury symulujące, które każdego programistę po chwili zastanowienia mogą zwalić z nóg, ale wygląd zewnętrzny, tzn. grafikę i muzykę, których trochę brakuje.

GYROS!



SUMMERCAMP

Gra ta otrzymała ostatnio bardzo wysokie noty w magazynach komputerowych na Zachodzie. Jest to bardzo rozbudowana gra platformowa, na którą składają się cztery duże etapy. Grafika w grze jest bardzo urozmaicona a bohaterowie narysowani są w stylu komiksowym. Na terenie obozu letniego (tytułowego SUMMERCAMP) na 24 godziny przed jego uroczystym otwarciem zaginęła obozowa flaga - jak wiadomo, ważny element każdego obozowiska.
W grze tej sterujemy sympatycznym gryzoniem - myszą o dźwięcznym imieniu Maximus. Jako stały bywalec terenu obozowiska poczuł się zobowiązanym do odszukania zaginionej flagi. Jak jednak przemierzać teren obozowiska, gdy ma się tylko kilka cetymetrów wzrostu?! Każda odległość wydaje się wtedy wielokilometrową wyprawą. Radą może być dla naszego bohatera np. samochód, ale skąd wziąć tak mały samochód? No cóż, należy go zbudować z różnych znalezionych na terenie obozu elementów.



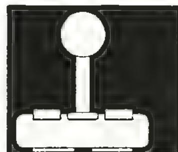
Każdy z etapów jest bardzo rozbudowany i trzeba się solidnie nabiegać, by znaleźć wszystkie ukryte wokół elementy samochodu. Na koniec każdej ze stref przychodzi nam ułożyć z zebranych elementów dany fragment samochodu - i należy się bardzo starać, bo gdy nie uda nam się to za pierwszym razem, będziemy zmuszeni powtarzać cały etap od początku! Na naszej drodze odwiedzimy wiele dziwnych miejsc (będziemy nawet na Księżycu!) i musimy się nieustannie spieszyć, bo otwarcie obozu tuż, tuż, a flaga ciągle się nie odnalazła...

SPACE CRUSADE

Na przełom marca i kwietnia znana firma Gremlin Graphics zapowiedziała wypuszczenie na rynek nowego hitu - Space Crusade. Jest to gra typu „board game" (jeżeli ktoś z Was grał wcześniej np. w Hero Quest'a, to już wie o co chodzi...), gdzie poszczególnymi postaciami kieruje się poprzez wydawanie im szczegółowych poleceń.
Akcja gry rozgrywa się tysiące lat licząc od dnia dzisiejszego, kiedy to ludzkość po wielu stuleciach niemrawego guzdrania się w otaczającej Ziemię przestrzeni kosmicznej w końcu odkryła zasadę działania napędu (warp engine), który w końcu pozwalał na pokonywanie gigantycznych odległości rzędu kilku tysięcy lat świetlnych w ciągu kilku godzin. Ludzkość kolejno kolonizowała „okoliczne" światy i wydawać by się mogło, że już nic przełomowego jej nie czeka. Jak wiadomo, dzięki podróżom swiat się kurczy, więc pewnego dnia imperium Ziemian zostało zaatakowane i prawie doszczętnie rozbite przez najeźdzców. Był to najdłuższy okres niedoli, jaki zaznała rasa ludzka. Po wielu pokoleniach rozbici na wiele grup Ziemianie zaczęli się jednoczyć i w końcu powoli udało się odzyskać dawną potęgę. Nie udało się jednak pobić przeciwnika, który nadal nękał ludzkość wojnami podjazdowymi, sabotażem i spiskami.
Do walki z tymi przejawami agresji powołano do istnienia specjalne oddziały antyterrorystyczne, zwane Space Marines. Ich członkowie byli najlepiej wyszkolonymi w walce ludźmi w Wszechświecie. W naszej grze jesteś właśnie dowódcą jednego z takich oddziałów. Do wykonania masz kilka zadań, z których każde musi być dokładnie zaplanowane (nad mapą), ustalony dobór sprzętu, zapasów, wyliczone w czasie. Marines, do dzieła!!

NIBBLE 92

A teraz dla urozmaicenia coś o nieskomplikowanej fabule - gra napisana przez „etatowych" twórców programów demonstracyjnych - Cosmos Designs. Jak wspomniałem wcześniej, gra nie jest skomplikowana w założeniach, co wcale nie znaczy, że łatwo jest ją ukończyć. Po naciśnięciu przycisku FIRE na naszym joystick'u rozpoczynamy grę jako... robak z tendencjami do szybkiego rośnięcia. Jako robaczek z sympatycznie uśmiechniętą buźką lubimy sobie dobrze podjeść. Wszystko zapewne obywało by się bez kłopotów, gdyby nie to, że wyłożone dla nas pożywienie znajduje się w labiryncie, który staję się dla nas coraz bardziej ciasny (w miarę jak my rośniemy od ciągłego jedzenia). Aby ukończyć każdy z etapów, należy zjeść wszystkie porcje pożywienia w danym labiryncie w ten sposób, aby niefortunnie nie... nadgryźć samego siebie! Zadanie początkowo jest niezbyt trudne, ale spróbujcie bezkolizyjnie zjeść wszystko począwszy od etapu np. czwartego! Jednym słowem, kilka chwil nie męczącej rozrywki zapewnione (a i zgłodnieć przy tej grze też można).



Listing nr 1

```
          ************************
          *   TAPE-BURGER-COPY!   *
          * (C) Commodore Kebab *
          *       by Polonus         *
          ************************
             lokacja:$0801-$0CFA
:0801  0B  08  C8  07  9E  32  30  36  (DA)
:0809  31  00  00  00  A9  4A  A0  08  (EB)
:0811  20  1E  AB  A9  0B  85  D3  A9  (7C)
:0819  74  A0  08  20  1E  AB  A9  80  (A4)
:0821  85  9D  A9  E0  85  FC  A9  00  (83)
:0829  85  FB  A8  A2  20  B1  FB  91  (57)
:0831  FB  C8  D0  F9  E6  FC  CA  D0  (84)
:0839  F4  A2  00  BD  88  08  9D  00  (90)
:0841  EF  E8  E0  FB  D0  F5  4C  83  (8E)
:0849  09  93  05  20  20  20  20  20  (4F)
:0851  12  54  41  50  45  2D  42  55  (F3)
:0859  52  47  45  52  92  20  54  48  (7E)
:0861  45  20  55  4C  54  49  4D  41  (9A)
:0869  54  45  20  43  4F  50  49  45  (4D)
:0871  52  0D  00  42  59  20  43  4F  (B7)
:0879  4D  4D  4F  44  4F  52  45  2D  (27)
:0881  4B  45  42  41  42  0D  00  20  (C0)
:0889  0B  EF  78  A9  38  85  01  4C  (23)
:0891  D1  DC  A2  05  86  AB  2C  F4  (90)
:0899  07  A2  04  B5  2A  95  AB  CA  (19)
:08A1  D0  F9  20  38  F8  2C  8F  F6  (24)
:08A9  20  98  EF  20  AC  EF  A5  B9  (8F)
:08B1  18  69  01  CA  20  CC  EF  A2  (CF)
:08B9  08  B9  AC  00  20  CC  EF  A2  (40)
:08C1  06  C8  C0  05  EA  D0  F2  A0  (C3)
:08C9  00  A2  04  B1  BB  C4  B7  90  (A5)
:08D1  03  A9  20  CA  20  CC  EF  A2  (B7)
:08D9  05  C8  C0  BB  D0  ED  A9  02  (EF)
:08E1  85  AB  20  AC  EF  98  20  CC  (4F)
:08E9  EF  84  D7  A2  07  EA  B1  AC  (CB)
:08F1  20  CC  EF  A2  03  E6  AC  D0  (AD)
:08F9  04  E6  AD  CA  CA  A5  AC  C5  (AC)
:0901  AE  A5  AD  E5  AF  90  E7  EA  (09)
:0909  A5  D7  20  CC  EF  A2  07  88  (DD)
:0911  D0  F6  C8  84  C0  58  18  A9  (FE)
:0919  00  8D  A0  02  4C  93  FC  A0  (F6)
:0921  00  84  C0  AD  11  D0  29  EF  (F2)
:0929  8D  11  D0  CA  D0  FD  88  D0  (AF)
:0931  FA  78  60  A0  00  A9  02  20  (C8)
:0939  CC  EF  A2  07  88  C0  09  D0  (D5)
:0941  F4  A2  05  C6  AB  D0  EE  98  (22)
:0949  20  CC  EF  A2  07  88  D0  F7  (1A)
:0951  CA  CA  60  85  BD  45  D7  85  (44)
:0959  D7  A9  08  85  A3  06  BD  A5  (5D)
:0961  01  29  F7  20  EE  EF  A2  11  (58)
:0969  EA  09  08  20  EE  EF  A2  0E  (24)
:0971  C6  A3  D0  E9  60  CA  D0  FD  (CE)
:0979  90  05  A2  0B  CA  D0  FD  85  (13)
:0981  01  60  A2  00  8A  9D  00  0E  (01)
:0989  E8  D0  FA  20  17  F8  90  01  (C3)
:0991  60  20  FC  0A  C9  00  F0  F3  (6B)
:0999  8D  00  0E  20  27  0B  99  01  (15)
:09A1  0E  C8  C0  C0  D0  F5  20  E6  (66)
:09A9  0A  A9  2F  85  00  A9  00  A0  (A5)
:09B1  0A  20  1E  AB  A2  00  BD  06  (8F)
:09B9  0E  20  D2  FF  E8  E0  10  90  (3A)
:09C1  F5  A9  0B  A0  0A  20  1E  AB  (CE)
:09C9  AE  01  0E  AD  02  0E  20  CD  (06)
:09D1  BD  A9  16  A0  0A  20  1E  AB  (C7)
:09D9  AE  03  0E  AD  04  0E  20  CD  (24)
:09E1  BD  A9  21  A0  0A  20  1E  AB  (F8)
:09E9  A9  00  85  C6  AE  00  0E  20  (0A)
:09F1  CD  BD  20  E4  FF  F0  FB  C9  (F1)
:09F9  03  F0  03  4C  48  0A  60  0D  (CA)
:0A01  46  49  4C  45  4E  41  4D  45  (2A)
:0A09  3A  00  0D  53  54  41  52  54  (C8)
:0A11  20  20  20  3A  00  0D  45  4E  (64)
:0A19  44  20  20  20  20  20  3A  00  (7D)
:0A21  0D  54  59  50  45  20  20  20  (24)
:0A29  20  3A  00  0D  0D  4C  4F  41  (35)
:0A31  44  20  53  54  41  54  55  53  (30)
:0A39  3A  00  20  4F  4B  21  00  20  (56)
:0A41  45  52  52  4F  52  21  00  A9  (0E)
:0A49  01  85  C3  A9  0F  85  C4  AD  (78)
:0A51  04  0E  38  ED  02  0E  85  AF  (50)
:0A59  AD  03  0E  38  ED  01  0E  B0  (A9)
:0A61  02  C6  AF  85  AE  A5  AF  18  (E7)
:0A69  65  C4  85  AF  A5  AE  18  65  (C8)
:0A71  C3  85  AE  90  02  E6  AF  20  (C9)
:0A79  A9  0A  B0  08  A5  BD  45  90  (7A)
:0A81  C9  00  F0  02  A9  01  8D  FF  (52)
:0A89  0E  48  A9  2F  85  00  A9  2C  (80)
:0A91  A0  0A  20  1E  AB  68  F0  06  (AE)
:0A99  A9  40  A0  0A  D0  04  A9  3B  (73)
:0AA1  A0  0A  20  1E  AB  4C  B0  0B  (7E)
:0AA9  A9  00  85  90  20  F9  0A  20  (E7)
:0AB1  27  0B  48  A5  01  29  F8  85  (4F)
:0AB9  01  68  91  C3  45  90  85  90  (2F)
:0AC1  A5  01  09  07  85  01  AD  01  (0B)
:0AC9  DC  10  28  E6  C3  D0  02  E6  (CC)
:0AD1  C4  A5  C3  C5  AE  A5  C4  E5  (0E)
:0AD9  AF  B0  03  4C  B0  0A  20  27  (EF)
:0AE1  0B  20  54  0B  C8  84  C0  58  (5E)
:0AE9  18  A9  00  8D  A0  02  20  93  (35)
:0AF1  FC  18  60  20  E6  0A  38  60  (09)
:0AF9  20  17  F8  20  54  0B  84  90  (BB)
:0B01  A9  07  8D  06  DD  A2  01  20  (A6)
:0B09  37  0B  26  BD  A5  BD  C9  02  (FD)
:0B11  D0  F5  A0  09  20  27  0B  C9  (F9)
:0B19  02  F0  F9  C4  BD  D0  E8  20  (EA)
:0B21  27  0B  88  D0  F6  60  A9  08  (2E)
:0B29  85  A3  20  37  0B  26  BD  C6  (B1)
:0B31  A3  D0  F7  A5  BD  60  A9  10  (08)
:0B39  2C  0D  DC  F0  FB  AD  0D  DD  (16)
:0B41  8E  07  DD  48  A9  19  8D  0F  (D5)
:0B49  DD  68  8D  20  D0  8D  18  D4  (CE)
:0B51  4A  4A  60  A0  00  84  C0  AD  (9A)
:0B59  11  D0  29  EF  8D  11  D0  CA  (73)
:0B61  D0  FD  88  D0  FA  78  60  0D  (C8)
:0B69  31  20  2D  4C  4F  41  44  20  (89)
:0B71  4E  45  58  54  20  50  52  4F  (E2)
:0B79  47  52  41  4D  00  0D  32  20  (12)
:0B81  2D  52  55  4E  20  50  52  4F  (CA)
:0B89  47  52  41  4D  0D  33  20  2D  (31)
:0B91  53  41  56  45  20  50  52  4F  (BD)
:0B99  47  52  41  4D  0D  0D  00  A9  (5D)
:0BA1  37  85  01  58  A9  00  8D  00  (78)
:0BA9  08  20  59  A6  4C  AE  A7  A9  (08)
:0BB1  0D  A2  28  20  D2  FF  A9  2D  (20)
:0BB9  CA  D0  F8  A9  68  A0  0B  20  (CF)
:0BC1  1E  AB  AD  FF  0E  D0  07  A9  (E2)
:0BC9  7E  A0  0B  20  1E  AB  20  E4  (CB)
:0BD1  FF  C9  03  D0  01  60  C9  31  (02)
:0BD9  D0  03  4C  0D  08  AE  FF  0E  (77)
:0BE1  D0  EC  C9  32  D0  11  A2  20  (9B)
:0BE9  BD  A0  0B  9D  60  01  CA  10  (72)
:0BF1  F7  20  1B  0C  4C  C5  DC  C9  (1A)
:0BF9  33  D0  D3  A2  0B  BD  0D  0C  (38)
```

```
:0C01  9D  60  01  CA  10  F7  20  1B  (67)
:0C09  0C  4C  CB  DC  A9  35  85  01  (C0)
:0C11  58  4C  00  EF  C6  01  58  4C  (75)
:0C19  B0  0B  78  A9  38  85  01  A2  (44)
:0C21  00  BD  00  08  9D  00  D8  BD  (A8)
:0C29  00  09  9D  00  D9  BD  00  0A  (19)
:0C31  9D  00  DA  BD  00  0B  9D  00  (E9)
:0C39  DB  BD  00  0C  9D  00  DC  BD  (C7)
:0C41  00  0E  9D  00  DE  E8  D0  D9  (7E)
:0C49  60  A2  00  BD  00  D8  9D  00  (48)
:0C51  08  BD  00  D9  9D  00  09  BD  (7B)
:0C59  00  DA  9D  00  0A  BD  00  DB  (68)
:0C61  9D  00  0B  BD  00  DC  9D  00  (92)
:0C69  0C  BD  00  DE  9D  00  0E  E8  (26)
:0C71  D0  D9  60  A2  04  BD  00  0E  (99)
:0C79  95  2A  CA  D0  F8  AE  00  0E  (68)
:0C81  CA  86  B9  A9  41  85  BB  A9  (FA)
:0C89  03  85  BC  A2  00  BD  01  0E  (43)
:0C91  9D  3C  03  E8  E0  C0  D0  F5  (93)
:0C99  A9  BB  85  B7  A0  00  84  20  (EB)
:0CA1  84  22  A9  08  85  21  A9  0F  (06)
:0CA9  85  23  B1  22  91  20  C8  D0  (A8)
:0CB1  F9  E6  23  E6  21  A5  23  C9  (43)
:0CB9  D8  D0  EF  A5  2D  85  AE  A5  (87)
:0CC1  2E  85  AF  60  20  74  DC  4C  (4E)
:0CC9  60  01  20  74  DC  4C  60  01  (23)
:0CD1  A9  2F  85  00  A0  00  84  20  (2F)
:0CD9  84  22  A9  D7  85  21  A9  D0  (82)
:0CE1  85  23  B1  22  91  20  C8  D0  (E0)
:0CE9  F9  C6  23  C6  21  A5  23  C9  (BB)
:0CF1  07  D0  EF  20  4A  DC  4C  15  (47)
:0CF9  0C  BD  BD  BD  BD  BD  BD  BD  (E8)
```

Listing nr 2

```
        ***********************
        * KEBAB-DIGI-DRUMMER! *
        * (C) Commodore Kebab *
        *     by Polonus        *
        ***********************
          lokacja:$0801-$10D6
:0801  0B  08  90  06  9E  32  30  34  (1E)
:0809  39  00  A0  00  78  E6  01  B9  (B5)
:0811  2E  10  99  FA  00  C8  D0  F7  (32)
:0819  4C  00  01  E8  A9  FF  9D  68  (E2)
:0821  F7  3E  2F  E0  18  90  F6  A9  (83)
:0829  0D  8D  F8  D1  A2  7E  4B  15  (57)
:0831  7F  1B  D0  78  A9  7F  CF  AA  (7E)
:0839  DC  EC  7B  0E  DC  82  70  1A  (D6)
:0841  FE  E8  1B  8D  8F  83  8D  12  (E4)
:0849  1E  93  EE  A0  3F  02  14  F9  (7A)
:0851  B3  15  4E  FC  FD  AF  F9  35  (92)
:0859  84  AE  A9  58  D3  87  4F  05  (36)
:0861  B8  06  DD  8C  07  90  12  11  (7D)
:0869  8D  0E  93  51  8D  0F  DD  92  (CD)
:0871  A2  64  A9  82  8D  92  F4  13  (57)
:0879  FE  A0  7B  61  03  8C  F0  F2  (2B)
:0881  58  CF  B8  E4  D1  49  2E  0E  (B4)
:0889  08  08  20  3E  3E  CB  C5  C2  (6C)
:0891  C1  C2  2D  C4  D2  D5  CD  CD  (90)
:0899  C5  D2  3C  3C  20  42  59  28  (89)
:08A1  FC  96  4F  4C  4F  4E  55  FD  (88)
:08A9  BA  9B  C2  CC  CF  CB  A1  2E  (BB)
:08B1  00  9B  D0  CF  DA  D9  C3  CA  (98)
:08B9  C1  D8  E2  23  F9  01  20  53  (BF)
:08C1  84  00  0D  00  4B  4C  41  57  (32)
:08C9  49  53  5A  45  3A  C6  31  82  (0F)
:08D1  EF  33  2C  20  C6  35  00  09  (96)
:08D9  02  08  0A  0F  0A  08  02  09  (05)
:08E1  00  F9  BE  7F  12  4B  2A  F9  (1B)
:08E9  2B  80  E8  6F  08  2E  EE  10  (CE)
:08F1  D0  E8  DC  F5  8D  16  95  BC  (39)
:08F9  9D  C5  98  40  56  BD  97  6B  (85)
:0901  41  81  2F  8A  60  A2  01  8E  (25)
:0909  4B  09  A2  06  BD  19  5C  27  (70)
:0911  CF  71  E0  BC  85  02  4F  AB  (81)
:0919  58  2C  20  26  CC  67  81  22  (C7)
:0921  56  0D  57  04  84  20  A9  12  (32)
:0929  85  21  98  AE  63  2E  CF  A2  (35)
:0931  02  67  21  57  60  20  7A  48  (FF)
:0939  B1  20  43  A6  A2  E5  A8  08  (F4)
:0941  C2  0C  0E  EC  E2  20  95  6C  (9B)
:0949  AF  C4  95  08  EE  BD  7A  BC  (B2)
:0951  11  78  5E  E8  E8  C8  9E  F4  (3F)
:0959  F0  EA  C6  02  4E  2F  8A  FD  (CE)
:0961  89  18  AA  4C  B9  09  48  20  (1C)
:0969  E1  AE  3D  58  00  4A  98  CF  (22)
:0971  C9  6F  E6  03  37  60  E9  3A  (61)
:0979  5E  00  BB  B3  70  C0  55  B7  (98)
:0981  D0  99  B0  B4  8A  99  A8  2C  (AC)
:0989  67  AD  DB  31  4A  5E  C9  39  (95)
:0991  20  D7  45  52  6C  04  8E  6D  (FD)
:0999  D3  4B  09  00  0A  85  04  8A  (E2)
:09A1  20  33  0A  05  04  D8  86  83  (48)
:09A9  9D  E9  30  60  69  09  60  B1  (9C)
:09B1  2D  85  02  A9  07  85  03  82  (01)
:09B9  65  B9  8C  04  AA  B9  8B  74  (62)
:09C1  6C  9A  23  0A  A6  02  9D  6B  (E8)
:09C9  A7  03  C8  E6  02  C6  03  D6  (E2)
:09D1  A5  60  20  E4  59  F5  01  85  (D9)
:09D9  BB  8D  A9  80  71  01  21  86  (04)
:09E1  3A  60  87  D0  04  B7  2C  D0  (CB)
:09E9  E8  C9  53  D0  F3  AE  F1  AC  (6F)
:09F1  E8  8A  29  07  8D  B1  8A  1B  (12)
:09F9  56  30  B0  02  38  45  3A  E6  (4C)
:0A01  47  B0  FA  C9  41  B0  04  C9  (8D)
:0A09  3A  B0  F2  18  60  13  14  16  (71)
:0A11  17  19  1A  1C  1D  1F  20  FA  (1D)
:0A19  ED  F5  25  26  00  4C  BE  0A  (4B)
:0A21  8E  D9  A9  13  4B  84  80  1E  (B1)
:0A29  FE  20  EE  C5  14  03  BD  A9  (38)
:0A31  8A  11  92  78  E0  1C  89  7C  (24)
:0A39  19  E7  E0  0D  90  90  48  CE  (96)
:0A41  C9  78  B0  E2  EE  58  3F  CE  (7B)
:0A49  9F  D0  CE  E8  D0  BE  AC  EB  (2C)
:0A51  14  AE  B7  48  66  04  9F  97  (37)
:0A59  AD  E7  F0  C5  CE  C4  5C  C8  (24)
:0A61  30  B1  13  D0  08  A2  00  8E  (DA)
:0A69  ED  D9  67  BB  0A  D1  44  90  (A7)
:0A71  86  D4  0A  AE  9A  9E  BA  CD  (B3)
:0A79  46  3D  0A  4C  E4  A2  84  DE  (5D)
:0A81  96  18  8D  3A  B1  84  20  92  (DD)
:0A89  09  20  61  0A  F0  FB  C9  0D  (A0)
:0A91  E0  6E  4C  B8  0A  C9  1D  D0  (4E)
:0A99  13  78  D6  E0  07  B0  E9  E8  (8A)
:0AA1  8E  02  D6  8A  0A  8D  32  09  (0D)
:0AA9  6C  89  9D  1E  AE  37  0B  F0  (FD)
:0AB1  D4  CA  4C  5E  B1  D6  11  D0  (EF)
:0AB9  28  FA  CB  C9  03  88  D3  EE  (D8)
:0AC1  0E  D0  C0  C9  91  49  A5  90  (6B)
:0AC9  B7  CE  0C  64  16  36  C9  93  (A3)
```

10...
20
30

```
:0AD1 D0 0F 20 6B 09 B2 5D A0 (B9)
:0AD9 1F 91 20 88 10 FB B3 10 (3B)
:0AE1 2B D0 0D 4A 08 C9 3F B0 (1C)
:0AE9 93 EE D5 51 C9 2D D0 0B (28)
:0AF1 AD 95 F0 84 CE 6A 48 9B (04)
:0AF9 38 DE 95 0A B0 C5 C9 34 (0B)
:0B01 B0 C1 48 74 AB AE 24 09 (95)
:0B09 F0 06 B5 26 28 F9 26 FA (3F)
:0B11 18 6D 23 09 AA 68 9D 78 (68)
:0B19 6F 31 B6 09 4C 46 0B 20 (A8)
:0B21 05 B3 4D 09 30 8D 73 04 (25)
:0B29 20 25 09 EE 19 D0 4C 31 (6A)
:0B31 EA 14 5F 01 E0 96 D0 09 (4B)
:0B39 B8 AD 65 68 60 10 17 91 (8E)
:0B41 6C 57 A9 20 EA B9 AD 34 (24)
:0B49 0C E4 8A A9 01 12 A5 A9 (A6)
:0B51 05 F0 04 CE 51 2D 60 A9 (10)
:0B59 05 8D 2C 0C AC 66 8D C0 (D2)
:0B61 20 B0 27 EE 15 B1 AE 29 (B2)
:0B69 03 2C 40 03 F0 E7 0A AA (4B)
:0B71 CA CA BD 60 0C A0 00 8C (ED)
:0B79 AA 7D 31 6C E9 F3 61 9C (31)
:0B81 FC 60 0D 0E 0E 10 10 11 (45)
:0B89 5A 6D 04 0C AE 84 14 EE (7E)
:0B91 DE BD 00 11 C9 FE D0 06 (F9)
:0B99 99 4D 06 0C 60 C9 FF D0 (28)
:0BA1 07 90 C5 8D 02 0C F0 DD (20)
:0BA9 29 3F 8D 03 0C AA A0 12 (36)
:0BB1 84 AF A9 9C 2F AE E0 00 (28)
:0BB9 D2 04 18 69 20 90 02 E6 (C8)
:0BC1 AF CA D0 F6 85 AE 4C 38 (D8)
:0BC9 0C 71 F9 1E 69 0F AD 61 (4F)
:0BD1 00 85 FD 58 00 4A 0B 61 (4E)
:0BD9 A9 D2 A8 44 85 FB AD 2E (DF)
:0BE1 3B C5 FC B0 12 A5 FD 29 (D0)
:0BE9 0F B5 97 D4 8D 20 D0 EE (23)
:0BF1 B4 0C D0 03 EE B5 0C A9 (C4)
:0BF9 AA 8D 18 03 A9 0C 8D 19 (54)
:0C01 03 AD 0D DD A5 FB FE 6D (7A)
:0C09 99 A2 0F FB F0 F5 37 0F (72)
:0C11 0E 6F 3F AC D5 C0 2F F0 (E8)
:0C19 3F 32 2F 57 BF 1A 76 C2 (52)
:0C21 FC 95 40 A5 10 14 9C 39 (7B)
:0C29 34 3A 96 30 87 E2 CE 03 (08)
:0C31 37 8D 65 55 EC 89 74 32 (9F)
:0C39 3A 6A B5 80 DD 54 13 43 (58)
:0C41 0E 45 BB DF 1C E3 F0 FE (F0)
:0C49 CA 74 20 3A 0A 48 E7 35 (2A)
:0C51 79 BC 69 02 61 75 ED CB (07)
:0C59 A9 87 65 44 33 21 11 5B (6F)
:0C61 00 00 01 11 11 22 23 33 (62)
:0C69 34 44 B6 A5 77 A6 0B 00 (6B)
:0C71 32 28 A9 4E 86 66 C4 75 (38)
:0C79 C5 A6 19 1B A4 3D 07 77 (D8)
:0C81 79 C4 56 DB 1F 3F 72 A4 (4F)
:0C89 03 00 7F 0F 2F AC 1F D3 (B5)
:0C91 00 B0 F0 00 83 B0 0E 0F (56)
:0C99 80 FC FF 28 14 F8 4F F0 (97)
:0CA1 0A 22 11 10 41 48 46 79 (15)
:0CA9 84 FE FF FC 90 50 50 B0 (82)
:0CB1 9A 80 07 78 DF 07 7B 58 (EE)
:0CB9 DA 8F 77 01 59 E8 10 2D (2B)
:0CC1 2C 00 08 9F 30 4B EF FB (A0)
:0CC9 1D B9 3C 85 BB 88 20 01 (EB)
:0CD1 50 08 BE 49 58 07 02 7D (73)
:0CD9 CE FF BD DA 60 9E 88 EA (EC)
:0CE1 9A F8 6B 01 77 00 28 95 (CF)
:0CE9 3E 16 D0 D7 CC EE 8D AF (0E)
:0CF1 8B F8 7A 68 FF F0 EB 07 (C6)
:0CF9 57 88 35 96 4A 83 10 54 (F7)
:0D01 AF FF E7 BA FD 27 93 1A (08)
:0D09 98 C9 13 28 38 52 17 78 (7E)
:0D11 FA 8E B4 57 55 99 EF FE (64)
:0D19 9C B1 32 48 39 68 74 95 (3B)
:0D21 BE 50 60 25 5B DF 7A DE (87)
:0D29 F8 E7 27 9B 8A 75 BD 2D (E0)
:0D31 56 82 08 36 A6 2F 5A C8 (96)
:0D39 8A AD EA CF FD A6 93 04 (1E)
:0D41 84 61 13 97 9D 96 EF 9C (27)
:0D49 B8 3F 64 8B BD A6 32 DD (BF)
:0D51 5C C6 9B 67 56 76 44 53 (99)
:0D59 68 A9 BD ED C6 26 65 58 (50)
:0D61 77 BE 68 99 C7 74 56 49 (3A)
:0D69 BD C6 BA 66 36 5B 79 AB (5C)
:0D71 9D C9 39 AB 57 34 75 9C (02)
:0D79 AB B8 77 8B 78 55 68 87 (98)
:0D81 DA 57 76 69 E0 F9 8E B9 (FC)
:0D89 FB 36 A3 77 97 78 7B A9 (2A)
:0D91 96 68 66 B7 16 C0 96 33 (B2)
:0D99 A9 F8 C5 6A 9A B9 CE A8 (70)
:0DA1 77 68 65 98 A9 75 0A AB (2D)
:0DA9 C7 44 9C 36 AD 9C 8B B9 (4F)
:0DB1 A7 66 A5 E2 90 9A 9A 2F (C2)
:0DB9 66 67 86 79 7A 56 87 7B (5F)
:0DC1 BD BB 6E B7 92 76 76 70 (7F)
:0DC9 08 9A 98 97 66 69 79 8A (49)
:0DD1 BD A9 3A 8A 89 53 B4 82 (5E)
:0DD9 B2 26 BA 97 D1 88 B8 78 (7B)
:0DE1 8A 54 6C B4 69 68 87 84 (82)
:0DE9 9A 9B 9A 96 6B 2A 98 39 (EF)
:0DF1 94 62 F6 AC 89 DA B3 91 (1E)
:0DF9 30 53 B3 64 4D 97 FE 27 (BA)
:0E01 0C 20 90 03 89 A5 15 6D (9D)
:0E09 A8 05 36 51 9A 8A E2 82 (2B)
:0E11 18 61 4F 3C 50 54 D4 D0 (AA)
:0E19 24 24 86 3F 65 03 32 27 (C2)
:0E21 F8 4C 92 90 97 98 FC 87 (54)
:0E29 E3 A7 C8 FC 99 10 12 67 (C3)
:0E31 99 D2 57 02 79 87 A5 A8 (D3)
:0E39 5B 75 B9 49 B5 5C B4 5B (50)
:0E41 B5 49 DA 44 8D B5 35 AE (16)
:0E49 B7 33 8C DB 63 37 AC C9 (B9)
:0E51 63 36 40 A0 2F 97 63 35 (40)
:0E59 31 3B C8 74 34 46 8A DE (94)
:0E61 BA 87 53 44 57 9C CD DC (16)
:0E69 A9 66 52 44 58 7B 8B F1 (E1)
:0E71 BB 99 75 65 52 45 68 8A (BF)
:0E79 AE DB BB B9 86 57 52 33 (7E)
:0E81 55 66 7A AA BB DC BB AB (0A)
:0E89 A8 77 94 BF 96 55 55 67 (5C)
:0E91 87 8A 75 CC CB AB AA 48 (A8)
:0E99 9F 66 54 44 54 65 56 88 (BA)
:0EA1 79 89 BA BB ED 6A BA A9 (CF)
:0EA9 A8 88 67 66 65 55 65 57 (AE)
:0EB1 75 67 98 79 C6 D7 A9 BB (0D)
:0EB9 AA AA 93 5A 87 68 67 76 (7A)
:0EC1 56 66 66 56 D0 6E 77 89 (A8)
:0EC9 89 A6 AA 9A AA A9 91 42 (61)
:0ED1 78 D1 5E 65 76 67 78 78 (67)
:0ED9 86 A8 53 05 71 9A 18 E8 (83)
:0EE1 93 83 76 F6 77 87 67 90 (90)
:0EE9 19 5C 50 98 88 89 99 98 (E5)
:0EF1 99 99 89 A9 5D 98 78 88 (F2)
:0EF9 87 87 88 78 77 77 B0 05 (29)
:0F01 98 88 FB 43 00 E9 2C 2D (C7)
:0F09 80 AB FF B8 1F 25 52 1E (72)
:0F11 A4 81 02 24 09 40 02 C5 (3F)
:0F19 59 11 35 28 01 B7 13 16 (66)
:0F21 9A A1 32 01 CA 03 0D C8 (45)
:0F29 04 01 02 02 B0 04 18 00 (7C)
:0F31 00 87 C8 C8 C8 A9 00 85 (CC)
:0F39 10 A6 FB 30 0D 06 FA 2A (E0)
```



```
:0F41 26 10 C6 FB 88 D0 F2 AA (4A)
:0F49 18 60 48 A1 FF 85 FA A2 (8B)
:0F51 07 86 FB E4 FF A6 FE D0 (45)
:0F59 02 C6 FF C6 FE 68 90 DD (49)
:0F61 E0 E7 D0 D9 A9 37 85 01 (34)
:0F69 58 4C 01 08 A9 10 E6 11 (0A)
:0F71 24 11 85 04 AA BC D8 07 (3F)
:0F79 20 24 07 A6 04 7D BA 07 (ED)
:0F81 48 A5 10 7D C7 07 A8 68 (2B)
:0F89 A6 02 D0 08 C0 00 D0 04 (62)
:0F91 C9 07 F0 D8 18 65 FC AA (B1)
:0F99 98 65 FD 85 49 A4 11 F0 (51)
:0FA1 20 8A 38 E5 11 B0 03 C6 (DA)
:0FA9 49 38 85 48 A5 FC E5 11 (0C)
:0FB1 B0 02 C6 FD 85 FC B1 48 (52)
:0FB9 88 91 FC C6 01 EE 20 D0 (77)
:0FC1 E6 01 98 D0 F1 60 28 07 (05)
:0FC9 0C 0E 29 00 08 18 38 78 (7B)
:0FD1 FF 7F 7F 00 10 20 40 80 (2A)
:0FD9 00 00 00 00 00 00 01 02 (FF)
:0FE1 00 00 00 00 00 01 03 07 (43)
:0FE9 0F 03 04 05 06 07 07 08 (E6)
:0FF1 09 04 04 05 06 07 09 0A (08)
:0FF9 0B 01 20 18 E5 B4 1A 86 (6C)
:1001 6B 29 84 D8 1B D9 FB 69 (7C)
:1009 F7 A9 BF 3E 21 98 7D 0C (97)
:1011 48 66 5B 1F 06 27 A9 2D (D1)
:1019 20 D2 FF 01 0B 41 DD A0 (B6)
:1021 DF C0 1E AB A2 3F B4 CD (8E)
:1029 12 CA 10 FA A8 87 FF 00 (62)
:1031 13 32 0F B9 2E 0F 99 1C (18)
:1039 07 C8 D0 F7 20 23 07 F0 (4F)
:1041 46 20 23 07 D0 30 20 22 (7C)
:1049 07 69 02 C9 04 90 27 D0 (61)
:1051 07 20 23 07 69 04 D0 1E (F2)
:1059 20 21 07 69 06 C9 0D D0 (33)
:1061 11 C8 20 21 07 69 0D C9 (32)
:1069 11 D0 07 A0 05 20 24 07 (CC)
:1071 69 1D EE 00 04 EA 85 11 (A9)
:1079 A6 FE A5 FF 20 8A 07 A5 (4B)
:1081 49 85 FF A5 48 85 FE 20 (ED)
:1089 23 07 85 02 F0 14 0A 2C (2F)
:1091 A9 03 85 11 20 21 07 A6 (EA)
:1099 02 D0 02 69 08 20 61 07 (BC)
:10A1 F0 9A 20 23 07 F0 E9 20 (E3)
:10A9 22 07 69 04 C9 06 90 E2 (45)
:10B1 D0 07 20 22 07 69 06 D0 (CA)
:10B9 D9 A0 05 20 24 07 69 0A (7E)
:10C1 C9 0C F0 0D C9 0D D0 CA (F1)
:10C9 A0 01 20 24 07 69 9E D0 (D6)
:10D1 C1 A0 07 D0 F5 BD BD BD (81)
```

Listing nr 3

```
10 REM < PRZEWIJANIE TEKSTU >
11 :
15 W=5:REM 5 WIERSZY OD GORY
16 X=1
17 :
20 TEKST$="TO JEST TEKST, KTORY BEDZIE SIE PRZESUWAL NA EKRANIE......    "
22 :
30 PRINT CHR$(147);:IF W=0 THEN 40
31 FOR I=1 TO W:PRINT:NEXT I
32 :
33 REM *** SCROLL ***
34 :
40 PRINT CHR$(32);CHR$(20);:POKE 211,39
50 X$=MID$(TEKST$,X,1):IF X$="" THEN X=1:GOTO 50
60 PRINT X$;CHR$(13);CHR$(145);CHR$(145);:X=X+1
70 FOR I=1 TO 60:NEXT I
80 GET A$:IF A$<>CHR$(32) THEN 40
85 :
90 END
```

Listing nr 4

```
***************************
*      Gadajaca Amiga     *
*                         *
* written by K.K./Quartet *
*     (c) 1992 Kebab      *
***************************

;Tekst zrodlowy dla assemblera SEKA
;> Exec.library
FindTask:          =        -294
AddPort:           =        -354
RemPort:           =        -360
OldOpenLibrary:    =        -408
CloseLibrary:      =        -414
OpenDevice:        =        -444
CloseDevice:       =        -450
DoIO:              =        -456
;> Translator.library
Translate:         =        -30
bsr         DoOpenDevice
tst.l       d0
bne         ErrorDev
move.l      4.w,a6
lea         TranslatorName(pc),a1
jsr         OldOpenLibrary(a6)
move.l      d0,TranslatorBase
```

```
tst.l        d0
beq           ErrorLib
move.l       TranslatorBase(pc),a6
lea           Phonems(pc),a1
lea           StartString(pc),a0
moveq        #[StopString-StartString],d0
move.l       #$200,d1
jsr           Translate(a6)

bsr           SayIt

move.l       4.w,a6
move.l       TranslatorBase(pc),a1
jsr           CloseLibrary(a6)

ErrorLib:
move.l       4.w,a6
lea           IoRequest(pc),a1
jsr           CloseDevice(a6)

ErrorDev:
move.l       4.w,a6
lea           DiskRep(pc),a1
jsr           RemPort(a6)

moveq         #0,d0
rts
*----------------------
SayIt:
lea           IoRequest(pc),a1
move.w       #$003,$1c(a1)
move.l       #Phonems,$28(a1)
move.l       #$200,$24(a1)
move.l       #$000,$2c(a1)
move.l       4.w,a6
jsr           DoIO(a6)
rts

DoOpenDevice:
move.l       4.w,a6
sub.l         a1,a1
jsr           FindTask(a6)

lea           DiskRep(pc),a0
move.l       d0,$10(a0)
lea           DiskRep(pc),a1
jsr           AddPort(a6)

lea           NarratorName(pc),a0
lea           IoRequest(pc),a1
move.l       #DiskRep,$0e(a1)
moveq        #0,d0
moveq        #0,d1
jsr           OpenDevice(a6)
rts

*----------------------
TranslatorBase:
dc.l          0
TranslatorName:
dc.b          'translator.library',0
NarratorName:
dc.b          'narrator.device',0,0
Phonems:
blk.b        $200,0

DiskRep:
blk.b        $20,0

IoRequest:
blk.b        $30,0
Rate:                    dc.w      150
Pitch:                   dc.w      110
Mode:                    dc.w      0
Sex:                     dc.w      0
                         dc.l      Chmasks
                         dc.w      4
Volume:                  dc.w      64
Frq:                     dc.w      22200
                         dc.l      0,0,0

Chmasks:
                         dc.b      1,1,1,1

StartString:
dc.b 'Kebab is Commodore computers'
dc.b ' magazine, just for you!',0

StopString:
```

Ogłoszenia drobne

Sprzedam tanio Commodore 128D, dyskietki, joysticki, FINAL II
Zbigniew Bohdanowicz
tel. 35-98-75 Warszawa

Sprzedam Action Replay MkII 1.600.000 zł (instr. w języku niemieckim)
Mariusz Piechówka
Box-98 74-100 Gryfino

Sprzedam niedrogo stacje LDW2000 super + 120 dyskietek + interface Centronics
Szymon Kukulski
ul. Piłsudskiego 28
34-500 Zakopane tel.44-40

Sprzedam Black Box V4.0 do Commodore 64 i 128; cena 150.000 zł.
Krzystof Stefanowicz
ul. Spokojna 41
72-518 Ładzin

Pilnie kupię kontroler + HD 40-80MB do A2000
Artur Biedacha
ul. Ks. Józefa 357/2
30-233 Kraków

C-64C, magnetofon, 600 gier, Final II, 2 joysticki, lektura w jęz. polskim z zakresu C-64 cena 2mln.
Piotr Sadzikowski
ul. Szafera 146/7
71-245 Szczecin

Sprzedam nowy "Vortex ATonce plus", Flicker-Fixer "Multivision 500", Scanner Alfa-Data, Amigę 500plus (2MB Chip-RAM), zewnętrzną stację HD (1.64 MB), sampler GVP Sound Studio, oryginalny Powerpacker 4.0, pcHandler, A64 (nowy emulator C-64
Kontakt listowny:
Silver Dream!'s
ul. Wojciechowskiego 28
71-476 Szczecin

Spotykamy się już po raz trzeci, niestety, znowu trochę spóźnieni. Nadrabianie dawnych opóźnień nie zawsze jest takie proste, a i o prawie Murphy'ego też nie sposób zapomnieć. Począwszy od tego numeru KEBABA, rozpoczynamy także serwis Public Domain (szczegóły - patrz poniżej) i coś nowego - pierwszy wydany przez nas program - Kebab Mon, czyli profesjonalny debugger dla Commodore 64.
W sprawie prenumerat - naprawdę BARDZO prosimy o umieszczanie na kwitach wpłaty swoich dokładnych adresów i nazwisk. Ich brak uniemożliwia nam realizacje zamówień! W związku z tym prosimy o kontakt następujących Panów: Eugeniusza Babulę z Leszna, Jana Jarmoszewicza z Gryfic, Andrzeja Ignaciuka z Poznania i Stefana Kuczkowskiego z Piły.
Często otrzymujemy zapytania o możliwość zakupienia za naszym pośrednictwem oryginalnych cartridge'ów (Action Replay 6 dla C-64 i Action Replay 3 dla Amigi). O tym, czy zdecydujemy się pośredniczenie w tych zakupach zadecyduje liczba osób deklarujących chęć zakupu. Postanowiliśmy więc, że osoby chcące zakupić któryś z w/w cartridge'ów, powinny potwierdzić ten fakt przez przelanie na nasze konto sumy 200000 zł do dnia 30 maja br. wraz z dokładnym adresem i nazwiskiem no i, oczywiście, informacją o typie interesującego je urządzenia. Po tym terminie albo odeślemy pieniądze (za mała ilość osób), albo wyślemy zamówiony sprzęt, za który Czytelnik dopłaci do pelnej ceny przy odbiorze. Dla informacji, ceny: AR 6 (C-64):120,-DM; AR 3 (Amiga): 199,-DM. Zapłata nastąpi w przeliczeniu złotówkowym, po aktualnym w momencie zapłaty kursie bankowym.
Korzystając z okazji chcielibyśmy podziękować wszystkim tym, którzy do nas napisali lub zadzwonili.
Redakcja.



Kupon ogłoszeniowy

----
imię i nazwisko

----
adres

treść:



COMMODORE KEBAB PUBLIC DOMAIN SERVICE:

Amiga Public Domain (dyskietka -20 tys.zł) - wersja demonstracyjna programu Art Department Pro V2.1; wiele przydatnych programów użytkowych, opisów (po polsku!) i listingi z Kebaba.

Commodore 64 Public Domain (dyskietka -15 tys.zł) - Mega Demo "Dutch Breeze" holenderskiej grupy Black Mail, listingi i inne programy (m.in. te zamieszczone w Kebabie).

Zakupu dokonać można przez przelanie wskazanej sumy na nasze konto i wysłaniem nam pokwitowania wpłaty wraz z dokładnym adresem i nazwiskiem zamawiającego.

profesjonalny, wielofunkcyjny debugger dla Commodore 64 autorstwa Pawła "Polonusa" Sołtysińskiego

KEBAB-MON V.5 (Designers' Version)

Pozwala w wygodny sposób zainstalować się w wybranym miejscu pamięci Commodore 64, a wiele przydatnych funkcji czyni z niego bezkonkurencyjne narzędzie pracy każdego programisty.
A oto kilka cech Kebab-Mon'a:
- zdolność do assemblacji i reassemblacji WSZYSTKICH rozkazów M6510 (w tym także wszystkich tzw. niepublikowanych);
- wygodny edytor ekranowy;
- możliwość interpretacji/edycji danych w pamięci jako znaki, graficzne definicje liter, sprite'y i sample(!);
- bezproblemowa praca w dowolnie skonfigurowanej pamięci (dostęp do całej pamięci C-64);
- współpraca z magnetofonem i stacją dysków (opcjonalnie kompresja danych przy SAVE i dekompresja przy LOAD, co pozwala na znaczne oszczędności nośnika podczas pracy);
- automatyczny relokator (!) kodu maszynowego;

...I wiele, wiele innych!

Cena programu wraz z nośnikiem (taśma lub dyskietka) i instrukcją : 50 tys. zł.

Wystarczy przelać na nasze konto podaną wyżej sumę i przesłać na adres redakcji pokwitowanie wpłaty wraz z dokładnym adresem i nazwiskiem zamawiającego.